NASA Technical Memorandum 4176

Neuromorphic Learning of Continuous-Valued Mappings From Noise-Corrupted Data

Application to Real-Time Adaptive Control

Terry Troudet
Sverdrup Technology, Inc.
Lewis Research Center Group
Cleveland, Ohio

Walter C. Merrill
Lewis Research Center
Cleveland, Ohio

NASA
National Aeronautics and Space Administration
Office of Management
Scientific and Technical Information Division

1990

Summary

The ability of feed-forward neural network architectures to learn continuous-valued mappings in the presence of noise was demonstrated in relation to parameter identification and real-time adaptive control applications. An error function was introduced to help optimize parameter values such as number of training iterations, observation time, sampling rate, and scaling of the control signal. The learning performance depended essentially on the degree of embodiment of the control law in the training data set and on the degree of uniformity of the probability distribution function of the data that are presented to the net during a training sequence. When a control law was corrupted by noise, the fluctuations of the training data biased the probability distribution function of the training data sequence. Only if the noise contamination is minimized and the degree of embodiment of the control law is maximized, can the neural net develop a good internal representation of the mapping and be used as a neurocontroller. A multilayer net was trained with back-error-propagation to control a cart-pole system for linear and nonlinear control laws in the presence of data processing noise and measurement noise. The neurocontroller exhibited noise-filtering properties and was found to operate more smoothly than the teacher in the presence of measurement noise.

Introduction

A major challenge for intelligent control (ref. 1) of complex Advanced Propulsion Systems (APS), such as the Space Shuttle Main Engine, is the real-time analysis of a massive amount of diverse sensor data. Such analysis can be used to directly perform low-level, real-time adaptive control, to diagnose faults, or to send real-time descriptions of the dynamic state of the APS to a high-level controller. In the first case, the low-level controller has to compute the control signal adaptively and in real time so that it can be applied to the controlled process for a given set of sensor data. In the second case, the sensor information must be translated in real time into one of several parameters which characterize specific failure modes of the APS. In the third case, the dynamic state needs to be identified in real time in order to allow an accurate health condition monitoring of the APS.

In most instances however, it is difficult, if not impossible, to derive realistic models of the physical phenomena and feedback mechanisms which govern the evolution of systems as intricately complex as APS, and the only information available often consists of experimental data collected in flight or during ground tests. Moreover, the presence of noise in such systems makes it even more difficult to extract the information contained in the experimental data and to perform accurate fault diagnosis and condition monitoring.

It is a major asset of neural networks to be able to extract features from finite sets of input and output data which are representative of arbitrary, unknown continuous-valued mappings (refs. 2 to 4). As arrays of simple computing elements, neural networks are easy to implement, and benefit from attractive real-time processing capabilities due to their massive parallelism (refs. 5 and 6). They store the extracted features in the distributed network of their interconnections, which gives them the fault tolerance desired in hostile or remote environments. Such cost-performance advantages make neural networks well fitted for the data processing of fault diagnosis and conditioning monitoring of the APS.

This report analyzes the ability of neural networks to learn continuous mappings and serve as parameter identifiers or real-time adaptive controllers when the data used for training have been corrupted by noise during sensor measurements and/or off-site data transmission. In the case of adaptive control, the noise incurred through sensor measurements corrupts the actual values of the state variables (as well as the control signal given by the actuator), and it alters the dynamic evolution that the controlled process would have had otherwise. This will be called plant or measurement noise. On the other hand, noise incurred during the (analog) processing of the sensor data only corrupts the description of the dynamic evolution of the controlled process. This will be called data processing noise and will be analyzed first for simplicity.

In the section **Training Architecture With Data Processing Noise**, a training architecture is proposed to analyze the possibility of learning from a teacher-controller in the presence of data processing noise. In the section **Controlled Cart-Pole System**, this training architecture is computer simulated on the cart-pole system for linear and nonlinear control laws. In both cases, the training sequences are analyzed in detail with noise-free and noise-corrupted training data. In the section **Example of Neuromorphic Learning of Nonlinear Control With Measurement Noise and Data Processing Noise**, the results are applied to the most general situation where the data representing the dynamics of the controlled process are corrupted with both types of noise.

Training Architecture With Data Processing Noise

In order to identify the factors and parameters which influence the neuromorphic learning of continuous-valued mappings from noise-corrupted data, it is simpler to consider first the effect of data processing noise. The continuous-valued mapping is chosen to be a control law: that is, a mapping from the space of state variables onto the space of control signals. As mentioned in the **Introduction**, such a mapping can be viewed from the perspective of identifying a collective parameter associated with several sensor data, or from the perspective of real-time adaptive control. Whereas the first case would apply to situations of fault diagnosis and component degradation, the second case would apply to situations where a neurocontroller would be used in place of a human teacher, a rule-based automated expert, or a natural servomechanism.

In the presence of data processing noise, the state $\vec{Z}$ of the controlled process and the applied control signal C which are transmitted to the neural net are

$$\left.\begin{aligned} \vec{Z} &= \vec{Z}_{ex} + \hat{n}_{\vec{Z}} \\ \text{and} \qquad C &= C_{ex} + \hat{n}_C \end{aligned}\right\} \tag{1}$$

where the noises $\hat{n}_{\vec{Z}}$ and $\hat{n}_C$ are simulated as independent, normally distributed, zero mean processes. For a high signal-to-noise ratio of the input signal; that is,

$$\frac{\max\|\vec{Z}_{ex}\|}{\sqrt{\text{var}(\hat{n}_{\vec{Z}})}} >> 1$$

the function ϕ that represents the control law $\phi(\vec{Z}_{ex}) = C_{ex}$ can be approximated, by using a Taylor expansion, as

$$\phi(\vec{Z}) \approx C_{ex} + \hat{n}_{\vec{Z}} \frac{\partial \phi}{\partial \vec{Z}}\bigg|(\vec{Z} = \vec{Z}_{ex}) \tag{2}$$

When the noise-to-signal ratio remains small, trying to learn the mapping ϕ from sets of noisy input and output vectors $(\vec{Z}, C)$ is equivalent to trying to learn ϕ from sets of input and output vectors $(\vec{Z}_{ex}, C_n)$ where only the control force is corrupted by the effective noise:

$$C_n = C_{ex} + \hat{n} \tag{3}$$

where $\hat{n}$ is the independent, normally distributed, zero mean process

$$\hat{n} \approx \hat{n}_C - \hat{n}_{\vec{Z}} \frac{\partial \phi}{\partial \vec{Z}}\bigg|(\vec{Z} = \vec{Z}_{ex}) \tag{4}$$

When the input and output data transmitted to the net are corrupted by noise, the factors and parameters which influence learning can be studied by analyzing the learning performance of the net as a function of the signal-to-noise ratio $\max|C_{ex}|/\sqrt{\text{var}(\hat{n})}$ of the output signal only.

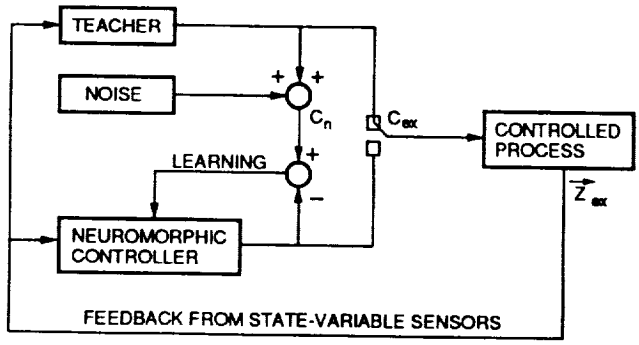


Figure 1.—Training architecture for neuromorphic learning with noisy data processing.

The first phase of the training consisted of sampling, at various times t_k, the state $\vec{Z}_{ex}(t_k)$ of the controlled process (fig. 1) and the control signal $C_n(t_k)$ given in equation (3). In the second phase, the set of training data $[\vec{Z}_{ex}(t_k), C_n(t_k)]$ was organized in input-output subsets before being applied to the neuromorphic controller as described below.

Controlled Cart-Pole System

The controlled process of figure 1 was chosen to be the cart-pole system (refs. 4, 7, and 8) represented in figure 2. Training data were recorded by placing the cart pole at arbitrary initial positions $[X(0), \theta(0)]$ with zero velocities and by driving it to the origin $(X = 0, \theta = 0)$ with a control force. While the cart pole was returned to its equilibrium position, the four-dimensional state vector $\vec{Z}(t) = [X(t), \dot{X}(t), \theta(t), \dot{\theta}(t)]$ and the control force $F(t)$ were regularly sampled over a certain period of time. Sampling rate and observation time were considered to be parameters of the training. In the first part of this section, a method based on the simple example of a

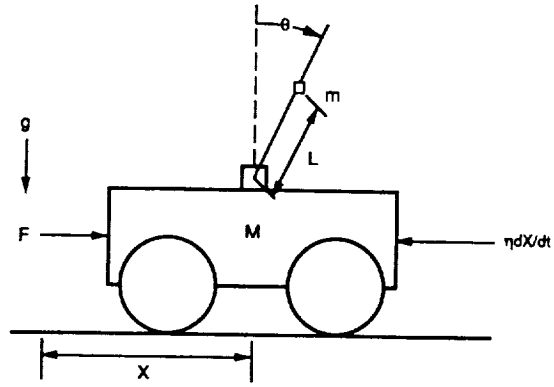


Figure 2.—Cart-pole system. The control force is applied to the cart in the presence of friction. Mass of cart, M, 1 kg; mass of pole, m, 0.1 kg; distance between base of pole and center of gravity of pole, L, 1 m; friction force applied to cart, η, 5 kg/sec; acceleration due to gravity, g, 9.81 m/sec^2.

linear control law is derived to optimize learning from noise-corrupted data. In the second part, it is applied and tested on a nonlinear control law.

Linear Control Law

For small deviations of the rod around its equilibrium position $\theta = 0$, the dynamics of the cart pole can be linearized and are given by

$$\left.\begin{aligned} \ddot{X} &= \frac{1}{M}\,[F(\vec{Z}) - f\dot{X}] \\ \text{and} \qquad \ddot{\theta} &= \frac{3}{4L}\,(g\theta - \ddot{X}) \end{aligned}\right\} \tag{5}$$

A linear controller that stabilizes this dynamic system to the origin ($X = 0$, $\theta = 0$) is (ref. 4)

$$F(\vec{Z}) = k_X X + k_{\dot{X}}\dot{X} + k_\theta\theta + k_{\dot{\theta}}\dot{\theta} \tag{6}$$

where the coefficients are $k_X = 11.01$, $k_{\dot{X}} = 19.68$, $k_\theta = 96.49$, and $k_{\dot{\theta}} = 35.57$.

Throughout this work, the term "neuron" is used to represent a simple processing element whose input and output response curve is a sigmoid that can be modeled as

$$\text{output} = \frac{1}{1 + \exp(-\text{input})} \tag{7}$$

Although the output of a neuron can take any value in the interval [0, 1], learning performance is known to be enhanced (ref. 9) when the asymptotes of the activation function given in equation (7) are eliminated by restricting the information domain of a neuron output to the interval [0.1, 0.9]. Thus, the neuromorphic learning of the continuous-valued mapping $F(\vec{Z})$ requires the scaling and offsetting of the last layer output o_n:

$$u = \frac{F}{F_0} = 2.5o_n - 1.25 \tag{8}$$

where F_0 is a constant parameter that normalizes the control signal over [−1, +1]. It is essential to emphasize that the choice of F_0 defines the domain where the mapping is to be learned, and it influences the neural computation as well. Equations (7) and (8) show that the net output cannot match values of F such that $|F| > 1.25F_0$ (practically, the net can only match accurately the domain $|F| \le F_0$). When a control law or a continuous-valued mapping is to be learned, it is imperative to first determine the range of variations, F_{max}, of the control signal before F_0 is chosen. The value of F_0 that satisfies $F_0 \ge |F_{max}|$ and corresponds to the best approximation of the continuous-valued mapping by the neural net can be obtained by minimizing an error function.

The neuromorphic architecture of figure 3 consists of a single neuron in the output layer and four fan-out units in the input layer. Each input unit feeds into the net one component of the state vector $\vec{Z} = (X, \dot{X}, \theta, \dot{\theta})$. Like the chemical voltage of a biological neuron is modulated by its synaptic connections before it contributes to the excitation or inhibition of another neuron, each signal of the input layer is subsequently modulated by a weight w and contributes to the total input signal of the output neuron:

$$i_{net} = w_X X + w_{\dot{X}}\dot{X} + w_\theta\theta + w_{\dot{\theta}}\dot{\theta} + w_{th} \tag{9}$$

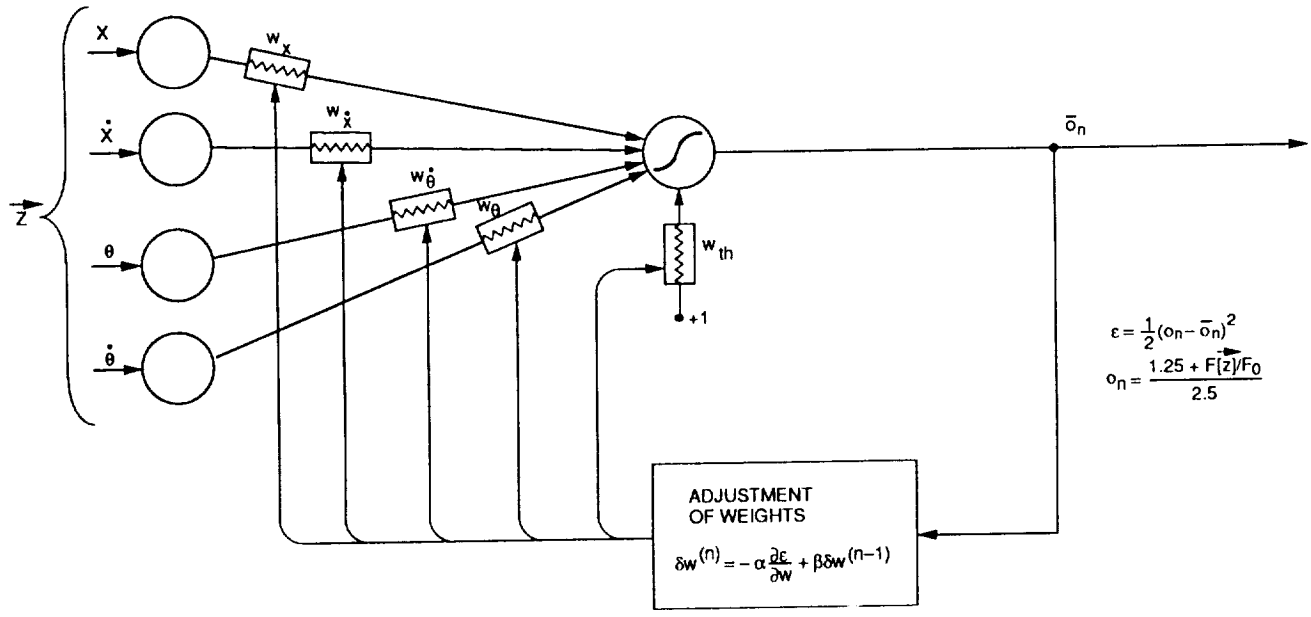


Figure 3.—Perceptron architecture to learn the linear control law $F = k_X X + k_{\dot{X}}\dot{X} + k_\theta\theta + k_{\dot{\theta}}\dot{\theta}$ from noise-free and noise-corrupted data; $\epsilon = \frac{1}{2}(o_n - \bar{o}_n)^2$; $o_n = |1.25 + F(\vec{Z})/F_0|/2.5$

where w_{th} represents the "synaptic" weight connecting the output neuron to an input neuron that is permanently "on" (i.e., with an output signal of +1). With the neuron activation function given in equation (7), w_{th} represents the threshold of external excitation below which the neuron is inhibited and above which it is activated.

Whether the architecture of figure 3 can learn the linear control law given in equation (6) is subject to the possibility of finding a set of weights w_X, $w_{\dot{X}}$, w_θ, $w_{\dot{\theta}}$, and w_{th} that satisfies the condition

$$\frac{1.25 + \dfrac{F}{F_0}}{2.5} = \frac{1}{1 + \exp(-i_{\text{net}})} \tag{10}$$

over the domain of variations of the control force F. For $|F|/F_0 < 1$, the Taylor expansion of equation (10) around the origin leads to the condition

$$i_{\text{net}} = 1.6\,\frac{F}{F_0} + O\left(\frac{F}{F_0}\right) \tag{11}$$

which is equivalent to

$$\left.\begin{aligned} & w_{th} = 0 \\ \text{and} \quad & w_\lambda = \frac{1.6}{F_0}\,k_\lambda + O\left(\frac{F}{F_0}\right) \quad \text{with} \quad \lambda = X,\ \dot{X},\ \theta,\ \dot{\theta} \end{aligned}\right\} \tag{12}$$

The above equations are formally satisfied over a finite domain of variations of the control force F only if the limit $F_0 \to \infty$, where $O(F/F_0) \to \infty$, is taken. However, as will be demonstrated in the next subsection, finite values of F_0 lead to a good approximation of the weights that map the pairs $[F(\vec{Z}), \vec{Z}]$ for $|F| < F_0$.

The results presented hereafter were obtained by implementing the back-error-propagation (BEP) algorithm on a VAX 8800 at the NASA Lewis Research Center. The initial values of weights and thresholds were chosen to be randomly distributed in the interval [-0.1, +0.1] to break symmetries that could eventually lead to spurious modes and bias the learning.

Learning from noise-free data.—In the absence of noise, learning is accomplished by training the network with a training data set. If the training data set covers only a limited region S_1 of the state space S_0 of the control law (i.e., $S_1 \subset S_0$), the neural net can, in general, only extract the input/output relation over S_1: it cannot generalize it over S_0. Consequently, the control law would be only partially embodied in the training data set S_1, and the neural net would only approximate the control law over S_1. In addition, the accuracy of the neural approximation of the control law over S_1 depends on how uniformly the available data can be ordered by magnitude throughout the dimensions of S_1. The ordering of the training data has to be uniform to prevent the net from focusing on any part of S_1 to the exclusion of the remainder.

One way to obtain a high degree of embodiment of the control law in the training data set is to observe many responses of the cart pole to random displacements from its equilibrium position. The degree of embodiment of the control law in the training data set grows (and subsequently converges) with the number of such motions N_{motions}, the length of observation time T, and the sampling rate f_s. The values of N_{motions}, T, and f_s, leading to a sufficiently representative training data set depend on the application, and, in general, have to be determined numerically. For this analysis, the pool of all cart-pole responses is called the training data set and consists of $N_{\text{motions}}Tf_s$ data points. Since the state of the cart pole was sampled at regular, fixed intervals until it returned to the origin, the distribution of the forces F used for training the net (as shown by a histogram of the number of occurrences versus magnitude, e.g.) was peaked around the origin. The distribution became more peaked as the observation time T increased. Clearly a random ordering of the training data would not be uniform and would bias the training to the origin, thus preventing the net from learning the control law for large displacements of the cart pole. Before training began, data were organized by dividing the interval [-1, +1] of the normalized force $u = F/F_0$ into N_{F_0} subintervals $I_k (k = 1, N_{F_0})$ of equal size. An approximately uniform ordering of the training data set could then be obtained from the random sampling of an interval I_k followed by the random sampling of the normalized force $u \in I_k$. Once selected, u and its corresponding state $\vec{Z}$ were then presented to the net as training data. Each $(u, \vec{Z})$ pair presented to the net represented the information required for one update, or training iteration, of the network weights. The sequence of all pairs presented to the net is called a training data sequence. The update procedure is given in the next paragraph.

For a randomly selected input state vector, the resulting network output $\bar{o}_n$ was compared with the target output $o_n = (u + 1.25)/2.5$. At each training iteration, the error $\varepsilon = \frac{1}{2}(o_n - \bar{o}_n)^2$ between the target output o_n and the network output $\bar{o}_n$ was back-propagated through all the net layers to update weights and thresholds by a steepest descent minimization of ε. The BEP update of the network was iterated until convergence. For a single-layer, feed-forward net, the changes $\delta w^{(n)}$ of the weights at the n^{th} iteration were

$$\delta w^{(n)} = -\alpha\,\frac{\delta \varepsilon}{\delta w} + \beta \delta w^{(n-1)} \tag{13}$$

In this equation, the first term is directly proportional to the gradient of the error, and the second term (or momentum term)

modulates the steepest descent update. For the perceptron of figure 3, weights and thresholds were updated by

$$\left.\begin{aligned}\delta w_{\vec{Z}}^{(n)} &= \alpha(o_n - \bar{o}_n)\bar{o}_n(1 - \bar{o}_n)\vec{Z} + \beta\delta w_{\vec{Z}}^{(n-1)} \\ \text{and} \\ \delta w_{th}^{(n)} &= \alpha(o_n - \bar{o}_n)\bar{o}_n(1 - \bar{o}_n) + \beta\delta w_{th}^{(n-1)}\end{aligned}\right\} \tag{14}$$

When mapping features are to be extracted, it is well known that any a priori knowledge, such as symmetries, can efficiently improve the net performance when it is incorporated explicitly in the neural computation. Since the control force changes as $F \rightarrow -F$ when the state vector changes as $\vec{Z} \rightarrow -\vec{Z}$, the ensemble of training data was chosen to be symmetric under the transformation $T^- = [(\vec{Z}, F) \rightarrow (-\vec{Z}, -F)]$ by randomly distributing the initial position and angle of the cart pole over the domain $D_{X\theta} = [-2 \text{ m}, +2 \text{ m}] \times [-20°, +20°]$.

Since all subintervals (I_k) were treated with equal probability during training, there had to be enough training data to provide as uniform a representation of the control law over each I_k as possible. In addition, the degree of embodiment of the control law in the training data set depended on the sampling rate f_s, which determined the degree of relatedness between the successively sampled data points $[F(\vec{Z}), \vec{Z}]$. To optimize F_s from an information-theoretic point of view, one can analyze the control law in the frequency domain. If f_c represents a cutoff frequency above which the spectral components of the control signal and the state vector are small (and can be neglected), f_s can be chosen to be equal to the Nyquist frequency, $f_{\text{Nyquist}} \approx 2f_c$. If $f_s << f_{\text{Nyquist}}$, information relative to the features between the state vector and the control signal will be missing in the training data set. If $f_s >> f_{\text{Nyquist}}$, the training data set will contain redundant information resulting in unnecessarily large memory space. The state space where the control law is to be learned will be bounded by the maximum value $F_{\max}$ of the control force needed to bring the cart pole back to the origin from an arbitrary position in $D_{X\theta}$ and with zero initial velocities. The value of $F_{\max}$ could be estimated from the training data set itself. For the parameter values of figure 1 and equation (6), the training data set was constructed by sampling 200 motions at the frequency rate f_s of 20 Hz, over a period T of 10 sec, leading to the estimate $F_{\max} \approx 60$ N.

Evaluation of learning performance without noise.—To evaluate the learning performance, we divided the interval $[-1, +1]$ of the normalized force $F/F_{\max}$, which represents the state space where the control law is to be learned, into $N_{F_{\max}}$ subintervals I_k ($k = 1, N_{F_{\max}}$) of equal size. After the net was trained with a given value of the parameter F_0, the accuracy of the neural approximation of the control law could be characterized by the total mean-squared error e^2

$$e^2 = \frac{\sum_{k=1}^{N_{F_{\max}}} e_k^2}{N_{F_{\max}}} \tag{15}$$

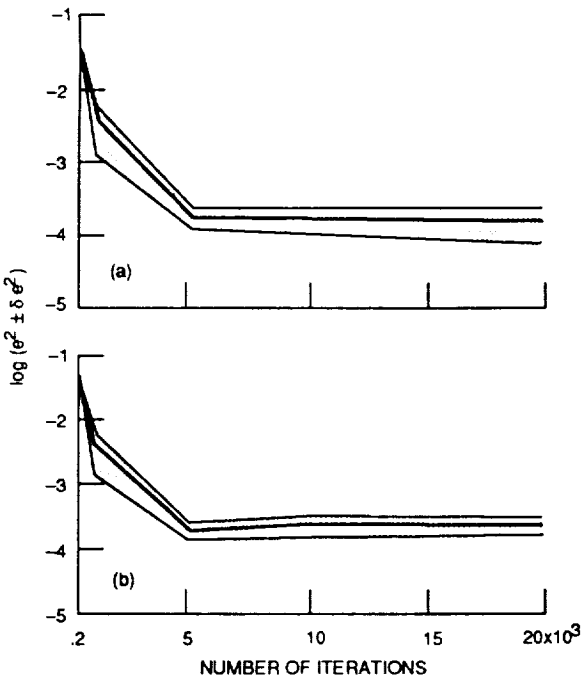
where e_k^2 is the mean-squared error over the subinterval I_k:

$$e_k^2 = \frac{\sum_{i=1}^{n(k)} \left[u(i)_{\text{net}} - u(i)_{ex}^{\text{target}}\right]^2}{n(k)} \left(\frac{F_0^2}{F_{\max}^2}\right) \tag{16}$$

In equation (16), $u(i)_{ex}^{\text{target}} = F_{ex}^{\text{target}}/F_0$ is one of the noise-free data values used for training and $u(i)_{\text{net}} = F_{\text{net}}/F_0$ is the output of the net corresponding to the same state variable. The error e_k^2 is averaged over the $n(k)$ normalized forces $F_{ex}^{\text{target}}/F_{\max}$ contained in I_k. This definition of the error makes it possible to compare the learning performances corresponding to different values of F_0 and to choose the optimal value of $F_0 \geq F_{\max}$ for which e^2 is minimal. The convergence of the algorithm can be estimated from the change in the error e^2 (eq. (15)) as the number of iterations increases. Here e^2 is obtained while an attempt is being made to reproduce the data used for training. Similarly, the accuracy of feature extraction can be estimated from the magnitude of the error e^2 obtained while trying to predict the control forces corresponding to state vectors that were not presented to the net during training. When both errors are small, the neural net has developed a good internal representation of the mapping.

Because of the statistical nature of the training, the internal representation of the mapping may vary from one training sequence to another. The reliability of the net to learn the same linear law from two or more training sets is therefore an important criterion of the neural computation and is called learning reliability. Learning reliability was tested by estimating the fluctuations of the total mean squared error e^2 (eq. (15)) over a set of 10 training sequences. For each training sequence, a "learning error" was estimated by calculating equation (15) over the full training data set. Similarly, a "generalizing error" was estimated by calculating equation (15) over a new training data set obtained by randomly generating 200 new motions of the cart pole sampled at the same rate and over the same period of time.

In Figure 4, parts (a) and (b) show the mean values and standard deviations of learning errors and generalizing errors, respectively, as functions of the number of update iterations. After 5000 iterations, the average values of the learning error and generalizing error are both small, with small standard deviation, indicating that iterative convergence has been reached. Figure 5 shows the generalizing error of the perceptron after 10 000 iterations for different values of the parameter F_0 (30, 60 ($= F_{\max}$), 120, and 180 N). To maintain a similar



(a) Error in learning mode calculated over the training data set.
(b) Error in generalizing mode calculated over the newly generated data that were not used to train the perceptron.

Figure 4.—Estimation of mean-squared error e^2 and its standard deviation, in training the perceptron to learn the linear control law from noise-free data. Ten training sequences; control signal normalizing factor, F_0, 60 N; steepest descent parameter, α, 0.2; momentum coefficient, β, 0.9.

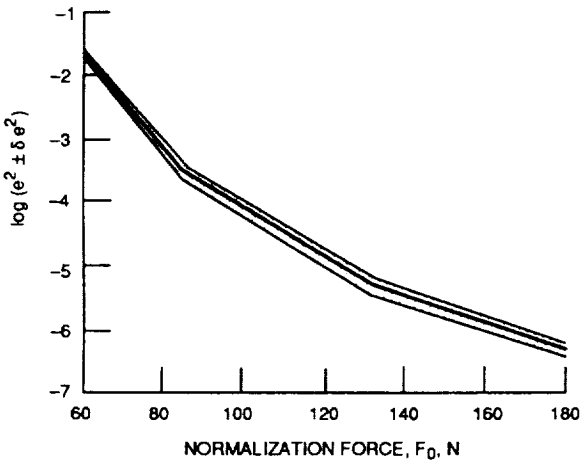


Figure 5.—Estimation of mean-squared error e^2 (generalizing mode) and its standard deviation versus control signal normalizing factor, F_0, after training the perceptron to learn the linear control law from noise-free data for several values of F_0. (Error calculated over newly generated data that were not used to train the perceptron.) Ten training sequences, 10 000 iterations per sequence; steepest descent parameter, α, 0.2; momentum coefficient, β, 0.9.

resolution of the ordering of the training data, we organized the interval $[-1, +1]$ of the normalized forces $u = F/F_0$ used for training into subintervals of equal size ($N_{F_0} = 7, 11, 21, 31$) corresponding to $F_0 = 30$, 60 ($= F_{max}$), 120, and 180 N, respectively. For $F_0 = 30$ N, the error was significantly higher than for $F_0 \geq F_{max} = 60$ N, since the net can only map the control law over a limited region of the state space: it is not able to generalize it on the remainder. As expected from equation (12), the error decreased as F_0 departed from F_{max} since the approximation of a linear control law by the perceptron improves as $F_0 \to \infty$. The rest of this subsection describes training for $F_0 = F_{max} = 60$ N.

Consequently, the learning performance of the net was tested in three different configurations. First, the learning open-loop configuration tested the ability of retrieving F from a state $\vec{Z}$ that was used for training (learning error). In this case, the net was essentially used as an analog memory. Second, the generalizing open-loop configuration tested the ability of the net to generate a control force from a state vector that was not used for training (generalizing error). In this case, the net was used as a parameter estimator. Third, the generalizing closed-loop configuration tested the ability of the net to stabilize the cart-pole process for a motion that was not used for training. In this case, the net was used as a real-time adaptive controller.

Convergence of the training sequence occurred after 10 000 training iterations, with $\alpha = 0.2$ and $\beta = 0.9$. In the learning open-loop mode, estimates of the total mean-squared error e^2 (eq. (15)) and its standard deviation over a random set of 10 training sequences were 0.00017 and 0.00008, respectively. In the generalizing open-loop mode, estimates of e^2 and its standard deviation were 0.00022 and 0.00008, respectively. In the generalizing closed-loop mode, estimates of e^2 and its standard deviation were 0.00019 and 0.00008, respectively. These results indicate the excellent performance of the neural net.

The dynamic characteristics of the cart pole controlled by the trained neural net were simulated for the initial state vectors $\vec{Z}(0) = (-0.7$ m, 0, $-17°$, 0) and $\vec{Z}(0) = (1.8$ m, 0, $-17°$, 0). As expected, the results obtained from the neuromorphic controller and the teacher were in perfect agreement.

Learning from noise-corrupted data.—Just as in the noise-free case, learning in the presence of noise was accomplished by constructing a representative training data sequence. However, the presence of noise limited the degree of representation that could be transferred from the original data into a training data sequence. This can be seen by studying the ordering process used to construct a uniform span of the state space as explained in the following paragraphs.

In the presence of noise, the normalized values of the control forces used as targets are no longer the exact values since

$$u_n^{\text{target}} = u_{ex} + \hat{n} \tag{17}$$

However, the mapping $F(\vec{Z})$ can still be learned by BEP if the training data set consisting of $u_n^{\text{target}}(i)$ is representative of the state space of the control law. Given the statistically averaged, error-squared function $\langle |F(\vec{Z}) + \hat{n}_F - G(\vec{Z})|^2 \rangle$ over the entire

state space, the BEP yields a function $G(\vec{Z})$ that minimizes the error as shown variationally in equation (18):

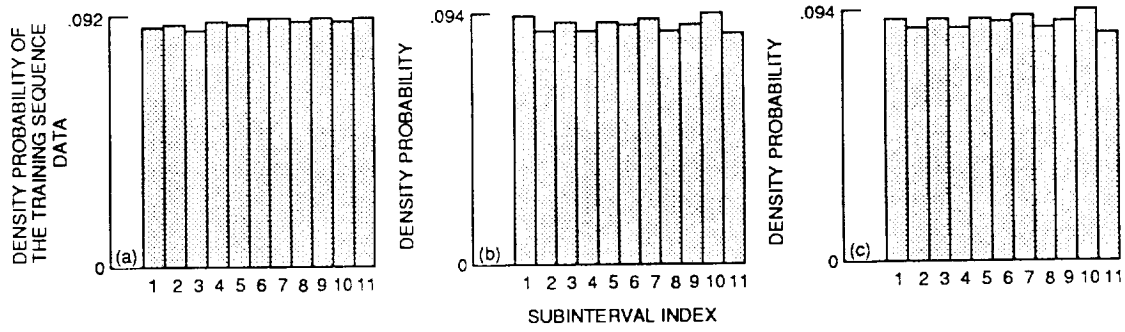
$$\frac{\delta\langle|F(\vec{Z}) + \hat{n}_F - G(\vec{Z})|^2\rangle}{\delta G(\vec{Z})} = 0 \Rightarrow G(\vec{Z}) = F(\vec{Z}) \qquad (18)$$

In the absence of noise, it is important to construct a uniformly distributed, training data sequence to increase the high degree of representation of the underlying function to be learned. This enables the net to reproduce the features of the function, rather than "memorize" the data, which is even more crucial in the presence of noise since the target data are not the exact values of the force. It is even more imperative not to learn any particular target data, but instead to minimize the error uniformly over the entire training data set.

Noise fluctuations tend to bias the learning through "data contamination" between the subintervals I_k. Because of noise, target data are likely to lie in subintervals that do not contain their exact data counterpart. This tendency implies that, unless the exact data are uniformly distributed throughout the state space (which in practice occurs rarely), the sampling by subintervals as described in the previous section will result in a less uniform distribution as the noise increases.

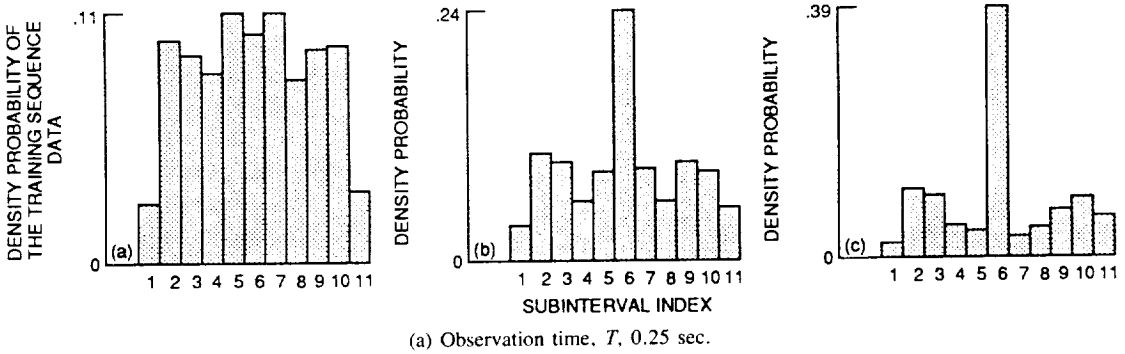
This phenomenon is illustrated in figures 6 to 8, which show the probability distribution function of the data actually presented to the net during training for various noise-to-signal ratios, $N/S = \sqrt{\text{var}(\hat{n})}/F_{\max}$, and for different observation times T of the cart-pole motions. As expected in the absence of noise, and because of the construction approach used, the data presented to the net during training (fig. 6) are, within a very good approximation, uniformly distributed throughout the state space. This uniformity does not depend on the length of observation (e.g., $T = 0.25$, 2.5, or 25 sec).

As shown in figure 7 in the presence of noise, the probability distribution function becomes less uniform as T increases. As noted earlier, the density distribution function of the forces $\vec{F}(Z)$ sampled from the teacher was more peaked around the origin as the observation time increased. In addition, the number of exact data that left their subinterval $I_k \in [-1, +1]$ because of the noise fluctuations was proportional to the number of data contained in I_k. As a result, the most highly populated subintervals increasingly populated their neighboring subintervals as T increased. Since the training data sequence was generated from the random sampling of a subinterval I_k followed by the random sampling of a $u \in I_k$, the probability distribution function of a training sequence was more peaked



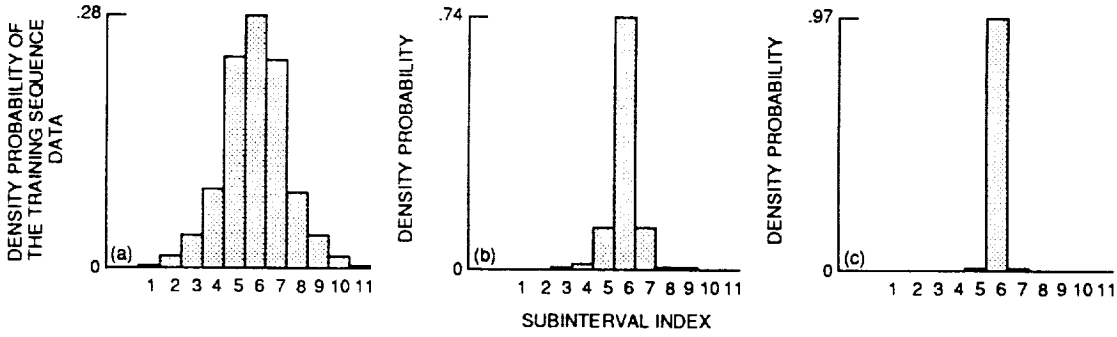
(a) Observation time, T, 0.25 sec.
(b) Observation time, T, 2.5 sec.
(c) Observation time, T, 25 sec.

Figure 6.—Probability distribution function of noise-free data presented to the network during training with the linear control law. (Index k defines the subinterval I_k of the state space [-1, +1] of the normalized force.)



(a) Observation time, T, 0.25 sec.
(b) Observation time, T, 2.5 sec.
(c) Observation time, T, 25 sec.

Figure 7.—Probability distribution function of data that are presented to the network during training with the linear control law; noise-to-signal ratio, N/S, 0.1. The index k defines the subinterval I_k of the state space [-1, +1] of the normalized force $F/F_{\max}$ of the linear control law.



(a) Observation time, T, 0.25 sec.
(b) Observation time, T, 2.5 sec.
(c) Observation time, T, 25 sec.

Figure 8.—Probability distribution function of data that are presented to the network during training with the linear control law; noise-to-signal ratio, N/S, 0.5. The index k defines the subinterval I_k of the state space [-1, +1] of the normalized force F/F_{max} of the linear control law.

as T increased. In figure 7, for $N/S = 0.1$, it is interesting to note that there is a relative depletion of the subintervals I_4, I_5, I_7, and I_8 surrounding I_6. Here, the small N/S ratio limited the spreading of the data of the overpopulated subinterval I_6 to its neighboring subintervals. Furthermore as N/S increased, the repopulation was no longer limited to a range of one or two subintervals, but it spread throughout the whole state space. In figure 8, for $N/S = 0.5$ and $T = 25$ sec, the probability distribution function of the data presented to the net during a training sequence reduced to that of the overpopulated subinterval I_6, which would clearly prevent effective learning.

As T becomes smaller, the probability distribution of a training data sequence becomes more uniform. However, as T becomes smaller, the control law is less embodied in the training data set since less of the state space is included in the data available for training. On the other hand, long observation times cause the noise repopulations to bias the training sequence by overemphasizing the contribution of the small amplitude motions of the cart pole: that is, $F \approx 0$. Consequently, an optimal value of T should be determined to minimize the noise contamination of the training data set and to maximize the degree of representation of the control law by the training data sequence.

Evaluation of learning performance in the presence of noise.—Prior to evaluating the learning performance with noisy training data, the boundaries of the (exact) state space where the control law is to be learned have to be estimated. In contrast to the noise-free case, where F_{max} can be obtained by direct observation of the training data set, an estimate of F_{max} is best obtained by training the net for several increasing values of F_0 and by subsequently analyzing the boundary changes of the state space spanned by the net output after it has been presented with the data used for training. An estimate of F_{max} is reached when the boundary of the state space of the net output does not change as F_0 increases. By using the same ordering technique as in the noise-free case, one can divide the estimated state space of the normalized control law F/F_{max} into $N_{F_{max}}$ subintervals of equal size. To optimize the parameters of the computation in the presence of noise, we introduced an error function to estimate the degree of accuracy to which the neural mapping approximated the exact unknown mapping

$$e^2_{noise} = \frac{\sum_{k=1}^{N_{F_{max}}} \left(e^2_{noise}\right)_k}{N_{F_{max}}} \tag{19}$$

where $\left(e^2_{noise}\right)_k$ is the mean-squared error over the subinterval I_k

$$\left(e^2_{noise}\right)_k = \frac{\sum_{i=1}^{m(k)} \left[u(i)_{net} - u(i)^{target}_{noise}\right]^2}{m(k)} \left(\frac{F_0^2}{F_{max}^2}\right) \tag{20}$$

In equation (20), $u(i)^{target}_{noise} = F^{target}_{noise}(i)/F_0$ is one of the noisy data values used for training, and $u(i)_{net} = F(i)_{net}/F_0$ is the output of the net corresponding to the same state variable. After training, however, the control force $F(i)_{net}$ output by the net is expected to be closer to the exact value than the (noisy) target value used for training. As a result, in contrast to the noise-free case, a better performance evaluation is expected by averaging the error over the $m(k)$ normalized forces $F(i)_{net}/F_{max}$ contained in I_k.

In the statistical limit where $m(k) \to \infty$ and for perfect learning, e^2_{noise} would be minimal and equal to $\text{var}(\hat{n})/F^2_{max}$. For this reason, we found it more convenient to grade the learning performance of the net by the quantity

$$\bar{e}^2_{noise} = e^2_{noise} - \frac{\text{var}(\hat{n})}{F^2_{max}} \tag{21}$$

This quantity goes to zero in the limit of perfect learning. Clearly, minimizing $\bar{e}^2_{\text{noise}}$ with respect to the parameters of the system is equivalent to minimizing e^2_{noise}.

It is essential to choose small values for the steepest descent parameter α in order to minimize the effect of the noise fluctuations. For a large α, samples with large deviations tend to overcontribute to the adjustments of the weights, and they mislead the search for the minimum. For a small α, the effect of such deviants tends to be balanced towards the average since samples with small deviations occur with a higher probability. From a geometric point of view, the fluctuations drastically complicate the topology of the error surfaces by creating more irregularities and increasing the possibility of the net getting trapped in local minima or flat spots. Small values of α favor adiabatic changes of the weights towards paths of the energy surface which correspond to averaged values of the training data: that is, $\langle u_n^{\text{target}} \rangle = \langle u_{ex} + \hat{n} \rangle = u_{ex}$. In the absence of noise (i.e., when the target values are the exact values), the momentum term speeds up the convergence process by amplifying the weight adjustments. In the presence of noise, a momentum term would amplify the undesired weight changes resulting from the highly deviant data, which would make adiabaticity more difficult to maintain. For this reason, the momentum coefficient β was chosen to be zero. The price to pay for a small steepest-descent coefficient and a zero momentum term is, of course, more iterations to reach convergence.

At each BEP iteration, the weights (including thresholds) were updated through equation (14). Like with noise-free data, the cutoff frequency f_c could be estimated from the spectral analysis of the training data set, since the addition of white noise amounts to a constant shift of the power spectra. For $N/S = 0.1$, the neural net was trained with a training data set of 15 000 randomly generated cart-pole motions sampled at $f_s = 20$ Hz over $T = 0.5$ sec. Figure 9 shows the noise-corrupted control force of an arbitrary cart-pole motion for this noise-to-signal ratio. For each training sequence, the BEP algorithm was run for 1 million iterations with the values of $\alpha = 0.006$, $\beta = 0$, and with the parameter $F_0 = F_{\max} = 60$ N. For 10 training sequences, estimates of $\bar{e}^2_{\text{noise}}$ (eq. (21)) and its standard deviation were 0.0037 and 0.0001, respectively.

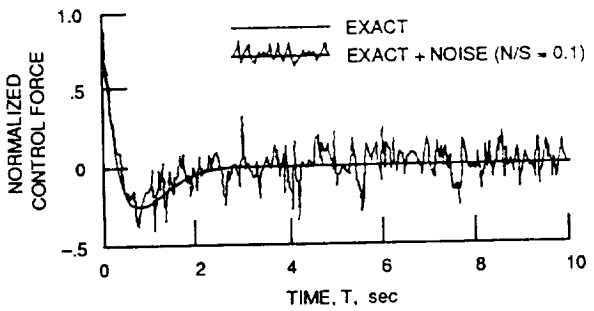


Figure 9.—Typical control forces applied by the linear teacher to stabilize the cart pole, and their noise-corrupted values used to train the network. Noise-to-signal ratio, N/S, 0.1; control signal normalizing factor, F_0, 60 N.

By direct comparison with the exact data, estimates of e^2 (eq. (15)) and its standard deviation were 0.0035 and 0.0001, respectively, in the generalizing open-loop mode. In the generalizing closed-loop mode, estimates of e^2 and its standard deviation were 0.0025 and 0.00006, respectively.

The dynamic characteristics of the cart pole controlled by the perceptron of figure 3, trained with noisy data are illustrated in figures 10 and 11 for $\vec{Z}(0) = (-0.45 \text{ m}, 0, -18°, 0)$ and $\vec{Z}(0) = (-1.9 \text{ m}, 0, 13°, 0)$, respectively. The agreement with the teacher is excellent.

For $N/S = 0.5$, the neural net was trained with a training data set of 35 000 randomly generated cart-pole motions sampled at $f_s = 20$ Hz during a shorter period, $T = 0.4$ sec. Figure 12 shows typical noise-corrupted data of the cart-pole motions. In each training sequence, the BEP algorithm was run for 10 million iterations with the values of $\alpha = 0.002$ and $\beta = 0$ and with $F_0 = 60$ N. For 10 training sequences,

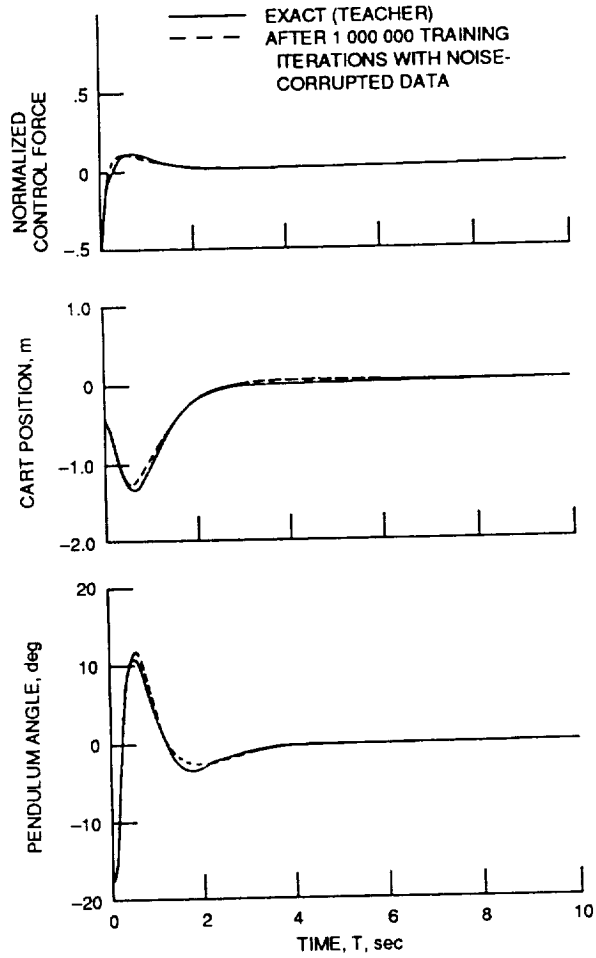


Figure 10.—Performance of the perceptron after 1 million training iterations with noise-corrupted data of the linear control law; initial state vector, $\vec{Z}(0) = (-0.45 \text{ m}, 0, -18°, 0)$. Noise-to-signal ratio, N/S, 0.1; control signal normalizing factor, F_0, 60 N; steepest descent parameter, α, 0.006; momentum coefficient, β, 0.

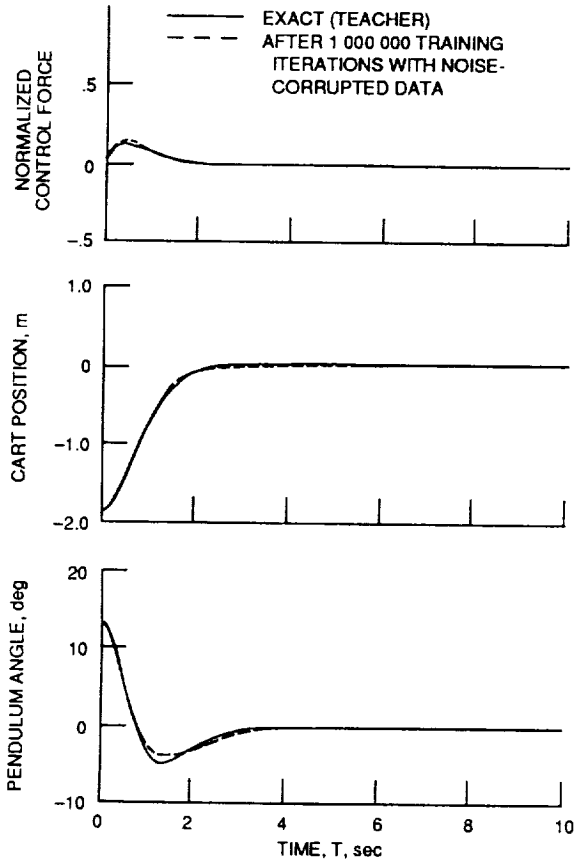


Figure 11.—Performance of the perceptron after 1 million training iterations with noise-corrupted data of the linear control law; initial state vector, $\vec{Z}(0)$ = (-1.9 m, 0, -13°, 0). Noise-to-signal ratio, N/S, 0.1; control signal normalizing factor, F_0, 60 N; steepest descent parameter, α, 0.006; momentum coefficient, β, 0.

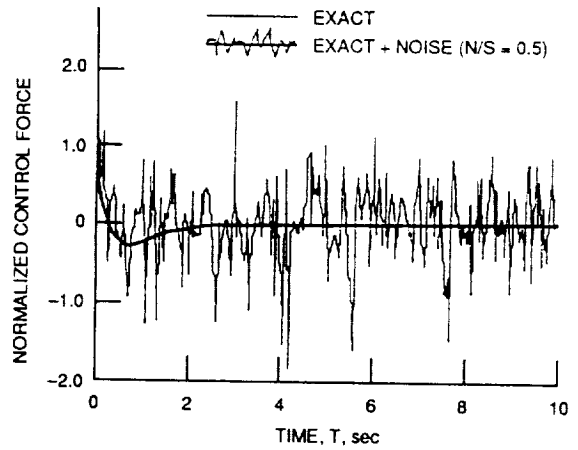


Figure 12.—Typical control forces applied by the linear teacher to stabilize the cart pole, and their noise-corrupted values used to train the network. Noise-to-signal ratio, N/S, 0.5; control signal normalizing factor, F_0, 60 N.

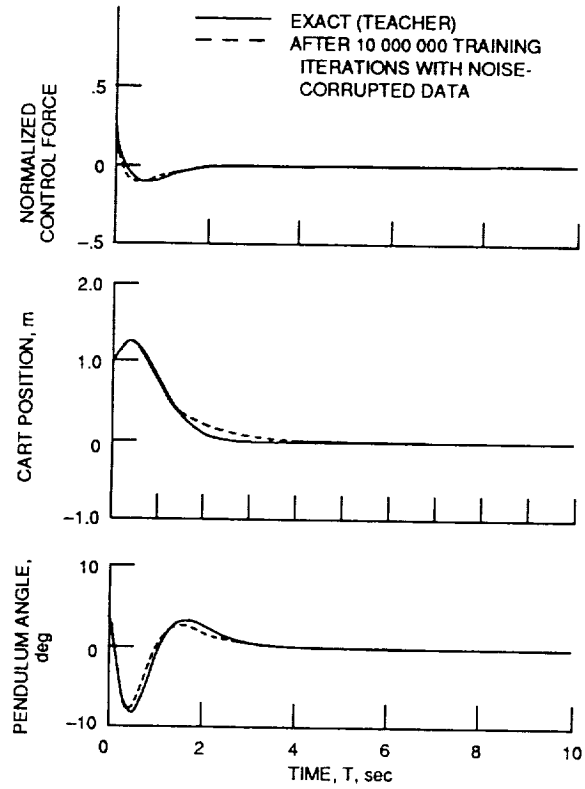


Figure 13.—Performance of perceptron after 10 million training iterations with noise-corrupted data of the linear control law; initial state vector, $\vec{Z}(0)$ = (1 m, 0, 3.8°, 0). Noise-to-signal ratio, N/S, 0.5; control signal normalizing factor, F_0, 60 N; steepest descent parameter, α, 0.002; momentum coefficient, β, 0.

estimates of $\bar{e}^2_{\text{noise}}$ and its standard deviation were 0.006 and 0.002, respectively. By direct comparison with the exact data, estimates of e^2 (eq. (15)) and its standard deviation were 0.006 and 0.0006 in the generalizing open-loop mode, respectively. In generalizing closed-loop mode, estimates of e^2 and its standard deviation were 0.004 and 0.0004, respectively. Typical curves for the cart pole controlled by the neural net are shown in figures 13 and 14. Even with such a high noise, the perceptron learned the process of returning the cart pole to the origin very well.

Nonlinear Control Law

The methodology developed for the linear controller was applied and tested on a nonlinear control law (ref. 4). The exact dynamical evolution of the cart pole (fig. 2) is given by the equations of motion

$$\ddot{\theta} = h_1 - h_2\ddot{X}$$

$$\ddot{X} = \frac{h_3 + F(\vec{Z})}{h_4}$$

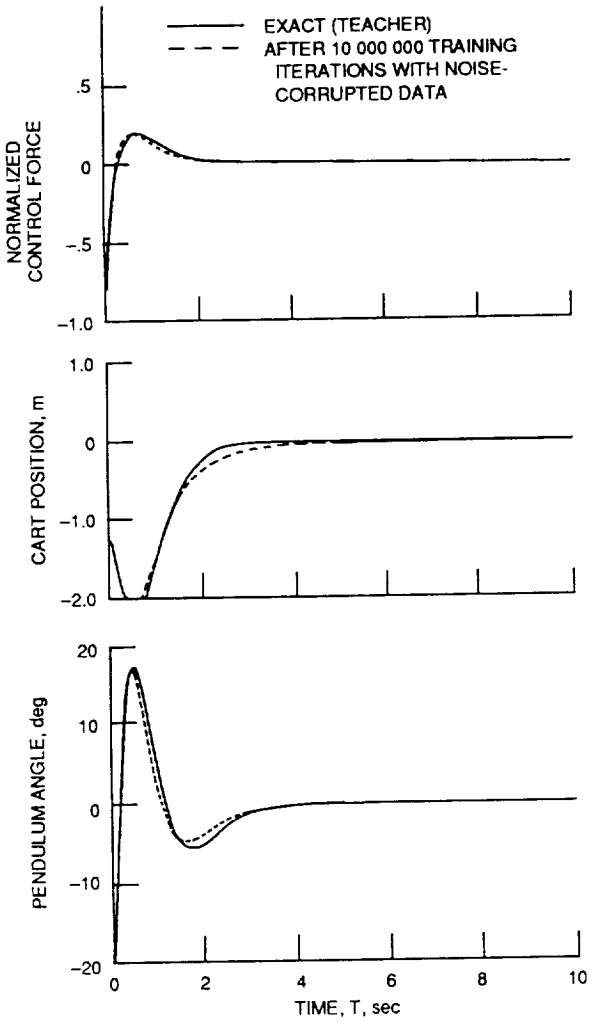


Figure 14.—Performance of the perceptron after 10 million training iterations with noise-corrupted data of the linear control law; initial state vector, $\vec{Z}(0)$ = (-1.25 m, 0, -19°, 0). Noise-to-signal ratio, N/S, 0.5; control signal normalizing factor, F_0, 60 N; steepest descent parameter, α, 0.006; momentum coefficient, β, 0.

where

$$h_1 = \frac{3}{4L} g \sin\theta$$

$$h_2 = \frac{3}{4L} \cos\theta \qquad (22)$$

$$h_3 = m(L \sin\theta\dot{\theta}^2 - \frac{3}{8} g \sin 2\theta) - f\dot{X}$$

$$h_4 = M + m(1 - \frac{3}{4}\cos^2\theta)$$

The control force $F(\vec{Z})$ was generated by applying a feedback linearizing and decoupling transform (ref. 10)

$$F(\vec{Z}) = \frac{h_4}{h_2}(h_1 + k_1\theta + k_2\dot{\theta} + c_1X + c_2\dot{X}) - h_3 \qquad (23)$$

where $k_1 = 25$, $k_2 = 10$, $c_1 = 1$, and $c_2 = 2.6$. Training data were generated by integrating equation (22) with the initial condition $\vec{Z}(0) = [X(0), 0, \theta(0), 0]$, where $X(0)$ and $\theta(0)$ were an arbitrary position and angle in the domain $D_{X\theta}$ extended to [-4 m, +4 m] × [-50°, +50°]. The neural architecture chosen to approximate the nonlinear control law $F(\vec{Z})$ was the feed-forward net represented in figure 15. The input layer, which had four linear neurons, fanned out the continuous values of the state variables $\vec{Z} = (X, \dot{X}, \theta, \dot{\theta})$ to the 16 neurons of the first hidden layer. Each neuron of the first hidden layer was connected to all four neurons of the second hidden layer. The neurons in the second hidden layer were all connected to a single neuron in the last layer. As shown in figure 15, each neuron input was connected to a fan-out unit that was permanently "on" (threshold term). The output signal of each neuron was modulated linearly by the (synaptic) weights before it excited or inhibited another connected neuron.

The layers were labeled by the index p from 0 to 4, $p = 0$ denoting the input layer. Layer p had $\nu(p)$ elements consisting of $[\nu(p) - 1]$ neurons and one fan-out unit which was permanently "on" and used to define the thresholds of the neurons of the $(p + 1)^{th}$ layer. The weight connecting the i^{th} neuron of the p^{th} layer to the j^{th} neuron of the $(p + 1)^{th}$ layer was represented by $w_{j,(p+1),i,p}$. The threshold of the j^{th} neuron of the $(p + 1)^{th}$ layer corresponded therefore to $w_{j,(p+1),\nu(p),p}$.

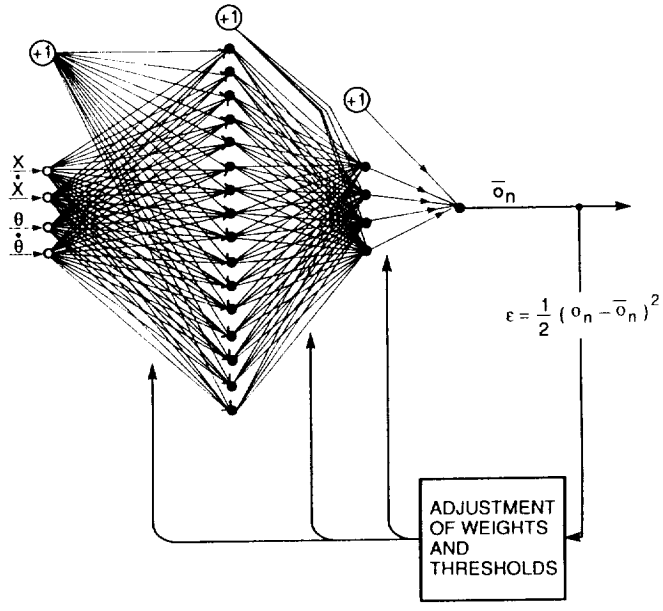


Figure 15.—Feed-forward neural network architecture with two hidden layers to learn the nonlinear control law (eq. (23)) from noise-free and noise-corrupted data ($o_n = |1.25 + F(\vec{Z})\,(+ \text{noise})\,/F_0|\,/2.5$).

At the n^{th} iteration, the weights were updated as

$$\delta w_{j,(p+1),i,p}^{(n)} = \alpha \bar{O}_{i,p} \Delta_{j,(p+1)} + \beta \delta w_{j,(p+1),i,p}^{(n-1)} \tag{24}$$

where the signal errors $\Delta_{k,(p+2)}$ at the $(p+2)^{th}$ layer were back propagated to the $(p+1)^{th}$ layer to give the signal error $\Delta_{j,(p+1)}$ at the $(p+1)^{th}$ layer

$$\Delta_{j,(p+1)} = \bar{o}_{j,(p+1)} \left[1 - \bar{o}_{j,(p+1)} \right] \times \left[\sum_{k=1}^{\nu(p+2)+1} \Delta_{k,(p+2)} w_{k,(p+2),j(p+1)} \right] \tag{25}$$

If $\bar{o}_n$ is the network output, and o_n the (noise-free or noise-corrupted) target output, the error signal $\Delta_{1,4}$ at the output layer is the gradient of the error given (as in eq. (14)) by

$$\Delta_{1,4} = (o_n - \bar{o}_n)\, \bar{o}_n (1 - \bar{o}_n) \tag{26}$$

In contrast to the linear case where perfect learning was obtained in the limit $F_0 \to \infty$, the only way to find the optimal value of F_0 for a nonlinear control law is to train the net for several values of F_0 larger than F_{max} and compare the learning performances.

Learning from noise-free data.—Owing to the symmetry of the problem, it is assumed that the control force $F(\vec{Z})$ changes as $F \to -F$ when the state vector changes as $\vec{Z} \to -\vec{Z}$. Therefore, the ensemble of training data was chosen to be symmetric under the transformation $T^- = [(\vec{Z}, F) \to (-\vec{Z}, -F)]$ by randomly distributing the initial position and angle of the cart pole over the domain $D_{X\theta} = [-4\text{ m}, +4\text{ m}] \times [-50°, +50°]$.

In the general case of a nonlinear control law, a spectrum analysis of the training data set provides only a gross estimate of the cutoff frequency. This value could be used as an educated guess for the optimization of f_s in minimizing the error e^2 (eq. (15)).

An upper bound for the control force needed to bring the cart pole back to the origin from a position in $D_{X\theta}$, and without initial velocities, is $F_{max} = 120$ N. In this section, training was performed over 200 motions sampled at $f_s = 20$ Hz during $T = 10$ sec. The steepest descent coefficient and the momentum term of the BEP algorithm were $\alpha = 0.2$ and $\beta = 0.9$. In learning the mapping $F(\vec{Z})$ (eq. (23)) and using it to control the cart pole, the neuromorphic controller was able to stabilize the pole to $\theta = 0$, but it would occasionally return the cart to an equilibrium position fluctuating in the vicinity of $X = 0$. To circumvent this numerical difficulty due to the existence of a local minimum or flat spot, we fine-tuned the controller by augmenting the training with data randomly sampled in the subinterval I_6, which is symmetric under the T^- transformation.

The mean-squared error was calculated for several values of $F_0 \geq F_{max}$ over a random set of 10 training sequences, which consisted of 99 000 gross-tuning iterations and 1000 fine-tuning iterations. Analyzing the error as a function of F_0 indicated that a minimum was reached for $F_0 = F_{max}$. With $F_0 = 120$ N, estimates of the total mean-squared error e^2 in the learning open-loop mode (eq. (15)) and its standard deviation were 0.0005 and 0.00017, respectively. In the generalizing open-loop mode, estimates of e^2 and its standard deviation were 0.00084 and 0.00037, respectively. In the generalizing closed-loop mode, estimates of e^2 and its standard deviation were 0.0007 and 0.00032. The results of the computation are shown in figures 16 and 17 for the initial state vectors $\vec{Z}(0) = (-1\text{ m}, 0, 45°, 0)$ and $\vec{Z}(0) = (3\text{ m}, 0,$

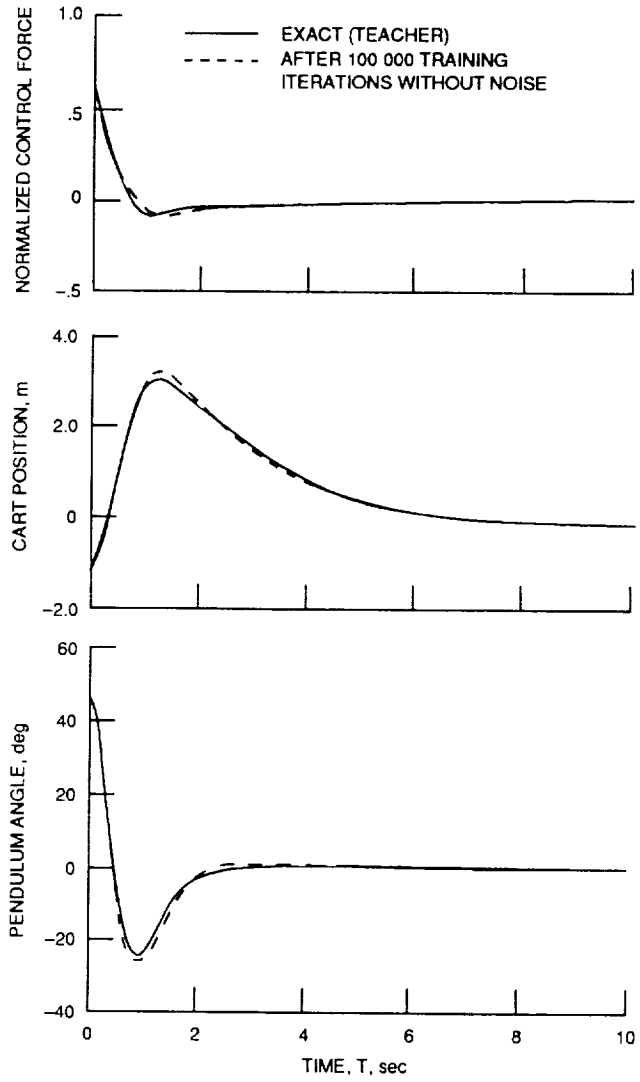


Figure 16.—Performance of the four-layer neural network after 100 000 training iterations with noise-free data of the nonlinear control law; initial state vector, $\vec{Z}(0) = (-1\text{ m}, 0, 45°, 0)$. Control signal normalizing factor, F_0, 120 N; steepest descent parameter, α, 0.2; momentum coefficient, β, 0.9.

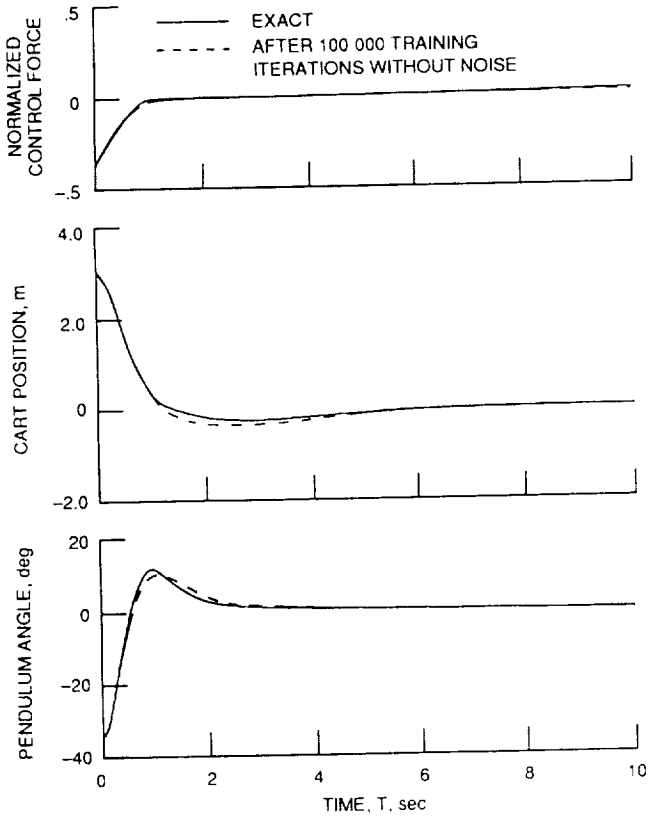


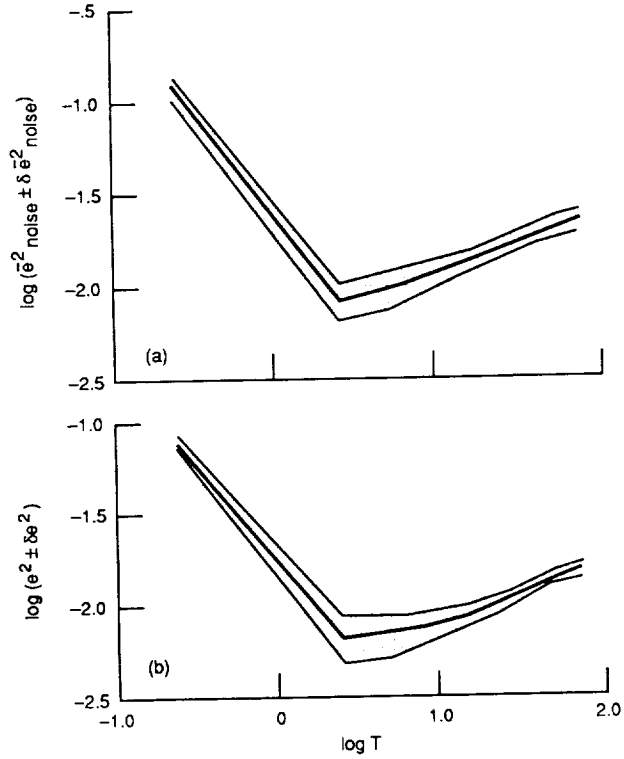
Figure 17.—Performance of four-layer neural network after 100 000 training iterations with noise-free data of the nonlinear control law; initial state vectors, $\bar{Z}(0)$ = (3 m, 0, −35°, 0). Control signal normalizing factor, F_0, 120 N; steepest descent parameter, α, 0.2; momentum coefficient, β, 0.9.

−35°, 0), respectively. The neural net was able to return the cart pole very satisfactorily to the origin from large angles and large displacements. This demonstrates that the internal representation that the net developed during training is a very good approximation of the mapping defined in equation (23).

Learning from noise-corrupted data.—The mapping to be extracted and learned by the net was assumed to be symmetric with respect to the T^- transformation, and the initial position and angle of the cart pole were also randomly distributed over $D_{X\theta}$. Towards the end of the training, the net was fine-tuned by training it with data sampled only from the subinterval I_6 symmetric with respect to T^-. Training data were generated by integrating the equations of motion (eq. (22)) and adding to the control force a noise normally distributed around zero.

The probability distribution functions of the data that were actually presented to the net of figure 15 during training were plotted for different observation times T and various noise-to-signal ratios. The characteristics of these probability densities are the same as those of the probability densities of the linear control law plotted in figures 6 to 8. For large values of T, the noise contamination of the sampled data prevented the construction of a uniformly distributed training data sequence. For small values of T, the data sampled from cart-pole motions were not sufficiently representative of the control law. Consequently, the mean-squared error e^2_{noise}(eq. (19)) was expected to be minimum for intermediate values of T. This is demonstrated in figure 18(a) for a noise-to-signal ratio N/S of 0.1. For comparison, the exact mean-squared error e^2 (eq. (15)) is plotted in figure 18(b) as a function of T. In figure 18, the errors were computed over a pool of data sampled from cart-pole motions over $T_{\max}$ = 100 sec, after training the net from data sampled from cart-pole motions over $T \le T_{\max}$. For larger noise-to-signal ratios, these "wells" would be more pronounced since the errors would significantly increase for larger values of T because of noise contamination.

With a noise-to-signal ratio of 0.1, training data were generated by sampling 1000 cart-pole motions at f_s = 20 Hz during T = 5 sec. With this set of parameters and following the approach described in the section ***Learning from noise-corrupted data***, an upper bound for the (exact) state space of the nonlinear control law was $F_{\max}$ = 120 N. The intensity of the noise is shown in figure 19. The convergence of the training is illustrated in figure 20, which shows $\bar{e}^2_{\text{noise}} \pm \delta\bar{e}^2_{\text{noise}}$ and $e^2 \pm \delta e^2$ as functions of the number of BEP iterations. A training sequence consisted of 490 000 gross-tuning iterations followed by 10 000 fine-tuning iterations, with the



(a) Mean-squared error $\bar{e}^2_{\text{noise}}$.
(b) Exact mean-squared error e^2.

Figure 18.—Estimations of mean-squared errors $\bar{e}^2_{\text{noise}}$ and e^2 and standard deviations after training the four-layer neural network to learn the nonlinear control law with a noise-to-signal ratio, N/S, of 0.1. The statistics refer to a set of five training sequences of 500 000 BEP iterations. Steepest descent parameter, α, 0.02; momentum coefficient, β, 0; control signal normalizing factor, F_0, 120 N.

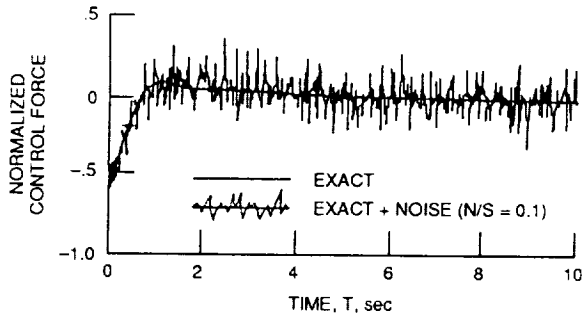
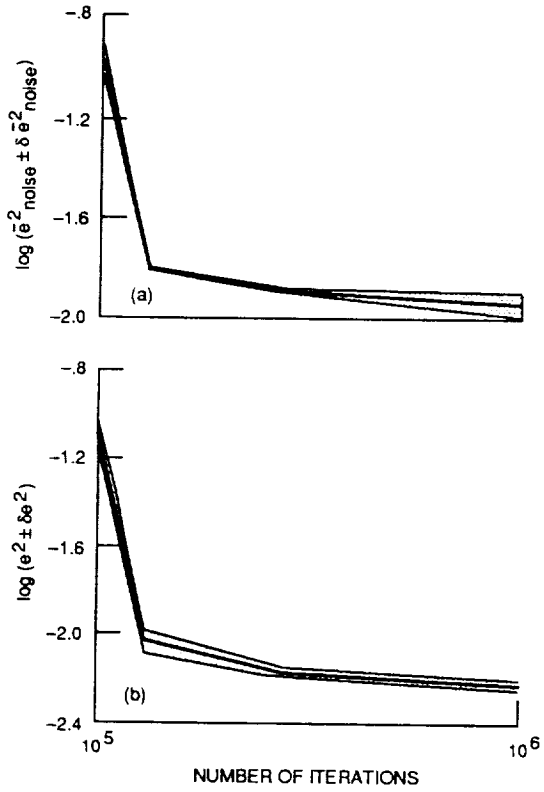


Figure 19.—Typical control forces applied by the nonlinear teacher to stabilize the cart pole and their noise-corrupted values used to train the neural network. Noise-to-signal ratio, N/S, 0.1; control signal normalizing factor, F_0, 120 N.



(a) Convergence of error due to noise as function of BEP iterations.
(b) Convergence of error as function of BEP iterations.

Figure 20.—Iterative convergence of the four-layer neural network in learning the nonlinear control law from data corrupted with a noise-to-signal ratio, N/S, of 0.1. (Same qualitative evolution of $\bar{e}^2_{\text{noise}}$ and e^2 with respect to the number of training iterations.)

parameter values $\alpha = 0.02$, $\beta = 0$, and $F_0 = F_{\max}$. For a set of five training sequences, estimates of $\bar{e}^2_{\text{noise}}$ and its standard deviation were 0.013 and 0.0003, respectively. By direct comparison with the exact data, estimates of e^2 (eq. (15)) and its standard deviation were 0.0068 and 0.0004 in the generalizing open-loop mode. The dynamic characteristics of the cart pole controlled by the net of figure 15 are shown in figures 21 and 22 for $\vec{Z}(0) = (3\text{ m}, 0, -35°, 0)$ and $\vec{Z}(0) = (-1\text{ m}, 0, 45°, 0)$, respectively.

For $N/S = 0.2$, 4000 cart-pole motions (fig. 23) were sampled at $f_s = 20$ Hz during $T = 2$ sec to train the net. The BEP parameters were $\alpha = 0.01$ and $\beta = 0$, and 950 000 gross-tuning and 50 000 fine-tuning iterations were used in each training sequence with $F_0 = F_{\max}$. For five training sequences, estimates of $\bar{e}^2_{\text{noise}}$ and its standard deviation were 0.016 and 0.001, respectively. By direct comparison with the exact data, estimates of e^2 (eq. (15)) and its standard deviation were 0.009 and 0.0008 in the generalizing open-loop mode. The dynamic characteristics of the cart pole controlled by the neural net are shown in figures 24 and 25 for $\vec{Z}(0) = (3\text{ m}, 0, -35°, 0)$ and $\vec{Z}(0) = (-1\text{ m}, 0, 45°, 0)$, respectively. For these two examples, the learning performance could be further enhanced by minimizing e^2_{noise} with respect to the scaling factor, $F_0 > F_{\max}$, and the sampling rate f_s.

These results show that, in spite of a significant amount of noise, the net was able to learn the control law (eq. (23)) within sufficient accuracy to return the cart pole satisfactorily to its equilibrium position.

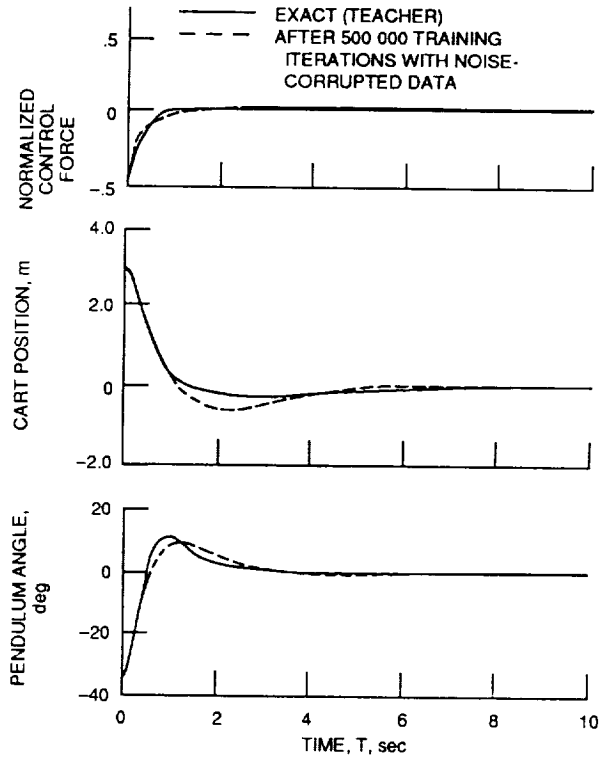


Figure 21.—Performance of the four-layer neural network after 500 000 training iterations with noise-corrupted data of the linear control law; initial state vector, $\vec{Z}(0) = (3\text{ m}, 0, -35°, 0)$. Noise-to-signal ratio, N/S, 0.1; control signal normalizing factor, F_0, 120 N; steepest descent parameter, α, 0.02; momentum coefficient, β, 0.

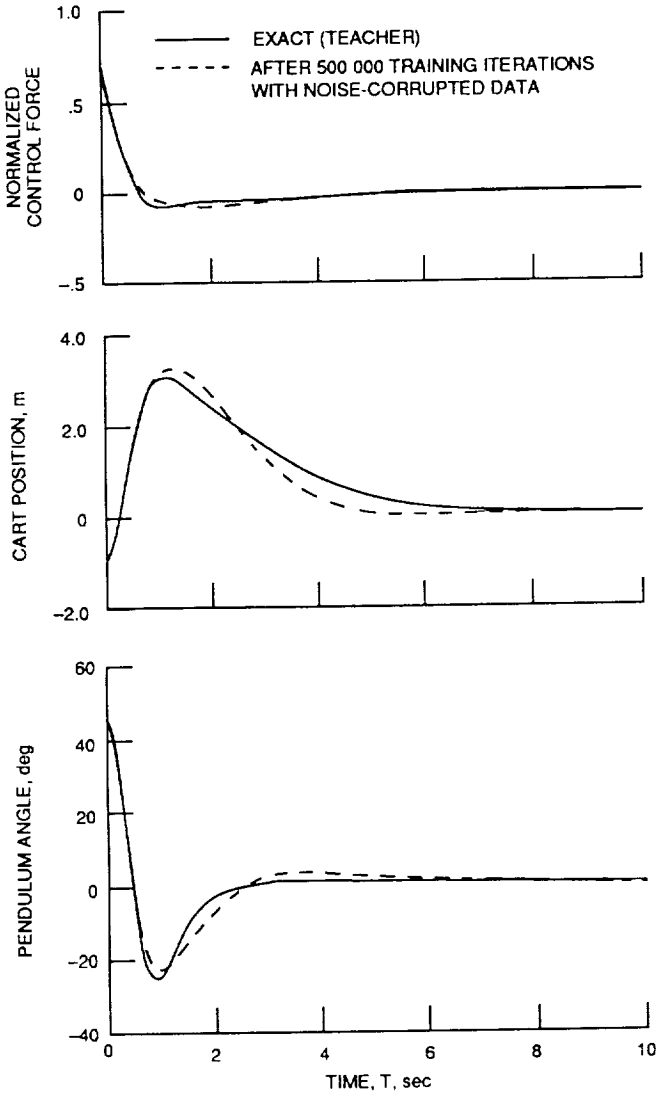


Figure 22.—Performance of the four-layer neural network after 500 000 training iterations with noise-corrupted data of the linear control law; initial state vector, $\vec{Z}(0)$ = (-1 m, 0, 45°, 0). Noise-to-signal ratio, N/S, 0.1; control signal normalizing factor, F_0, 120 N; steepest descent parameter, α, 0.02; momentum coefficient, β, 0.

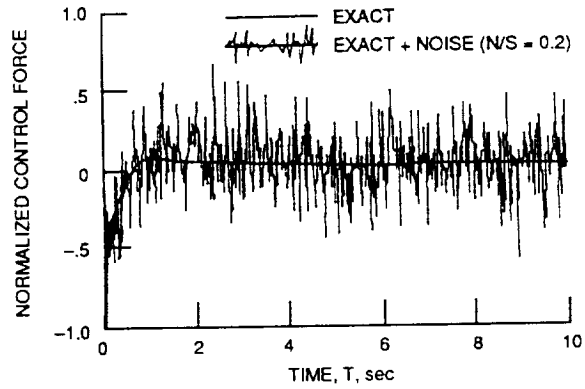


Figure 23.—Typical control forces applied by the nonlinear teacher to stabilize the cart pole, and their noise-corrupted values used to train the neural network. Noise-to-signal ratio, N/S, 0.2; control signal normalizing factor, F_0, 120 N.

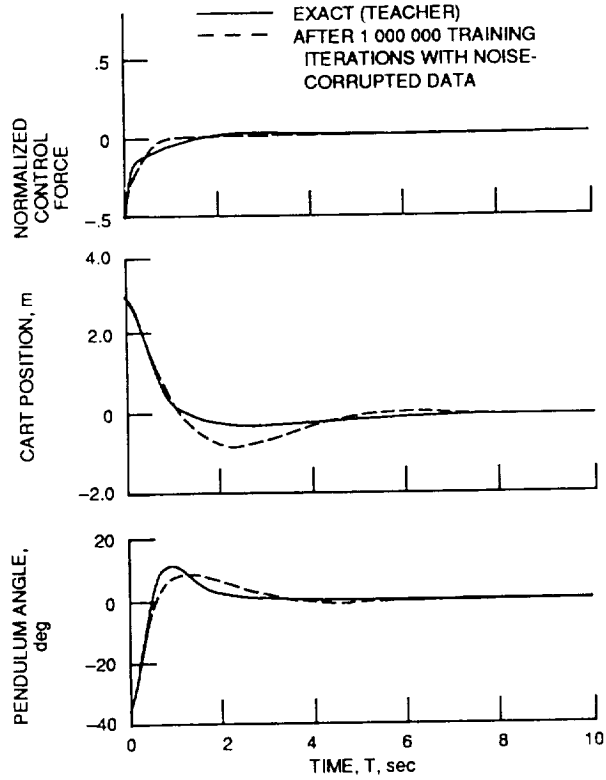


Figure 24.—Performance of four-layer neural network after 1 million training iterations with noise-corrupted data of the nonlinear control law; initial state vector, $\vec{Z}(0)$ = (3 m, 0, −35°, 0). Noise-to-signal ratio, N/S, 0.2; control signal normalizing factor, F_0, 120 N; steepest descent parameter, α, 0.01; momentum coefficient, β, 0.

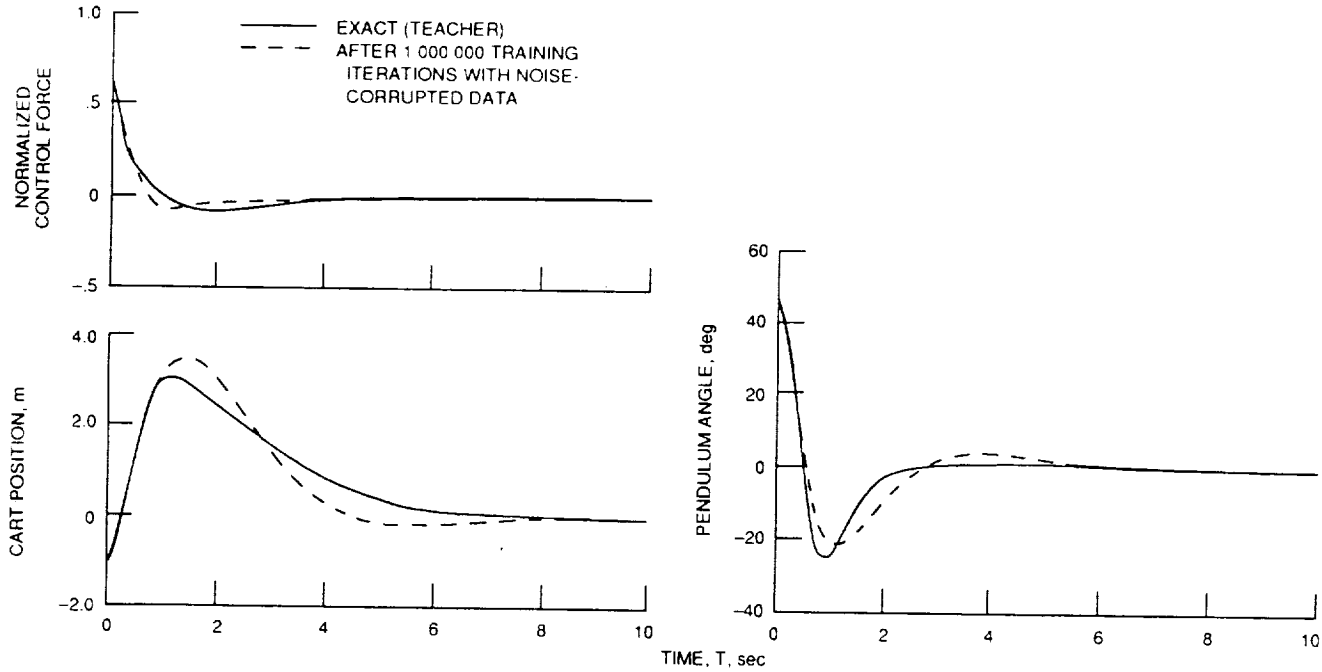


Figure 25.—Performance of four-layer neural net after 1 million training iterations with noise-corrupted data of the nonlinear control law; initial state vector, $\vec{Z}(0)$ = (-1 m, 0, 45°, 0). Noise-to-signal ratio, N/S, 0.2; control signal normalizing factor, F_0, 120 N; steepest descent parameter, α, 0.01; momentum coefficient, β, 0.

Example of Neuromorphic Learning of Nonlinear Control With Measurement Noise and Data Processing Noise

The ability of the feed-forward net to map a nonlinear control law was tested further by introducing the training architecture shown in figure 26. Here the dynamics of the cart pole controlled by the teacher were corrupted by noise. In addition, the representation of the cart-pole motions transmitted to the net was corrupted because of a noisy data communication link.

In the first phase of collecting the training data, cart-pole motions were generated from various initial positions randomly distributed over $D_{X\theta}$ = [-4 m, +4 m] × [-50°, +50°]. If the four-dimensional vector $n_{\vec{Z}}^{s}$ represents the noises associated with the measurement of the actual state vector $\vec{Z}^a$, the value of the state vector $\vec{Z}^s$ passed to the teacher is

$$\vec{Z}^s = \vec{Z}^a + \hat{n}_{\vec{Z}}^{s} \tag{27}$$

Although the teacher knows the exact transfer function $\vec{F}[Z(t)]$, noise may occur during the physical application of the force and create a discrepancy between the force applied by the actuator, $\vec{F}^a$ and the desired force $\vec{F}(\vec{Z})$,

$$\vec{F}^a = \vec{F}(\vec{Z}) + \hat{n}_F^a \tag{28}$$

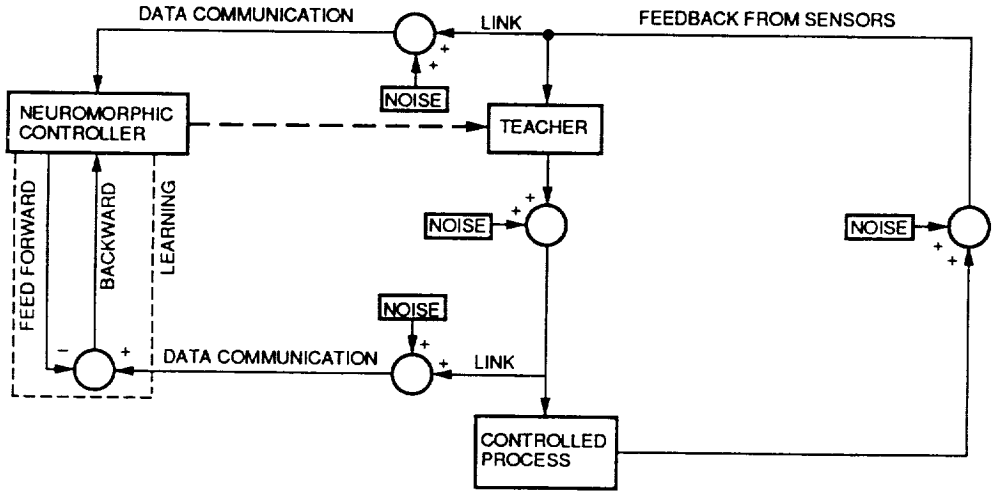


Figure 26.—General training architecture for neuromorphic learning with noisy sensor measurements (noise within control loop) and noisy data processing (noise on the data communication links).

In contrast to the training architecture of figure 1, for the architecture of figure 26 noise was included within the control loop itself. As a result, for the same initial conditions, the architectures of figures 1 and 26 led to different motions of the cart pole. In this numerical application, an N/S of 0.02 was chosen for the fluctuations of $\hat{n}_F^a$, $\hat{n}_X^s$, $\hat{n}_{\dot{X}}^s$, $\hat{n}_\theta^s$, and $\hat{n}_{\dot{\theta}}^s$. Figure 27(a) shows the normalized control force actually applied by the teacher-controller to the cart pole for the initial state vector $\vec{Z}(0) = (-1\text{ m}, 0, 45°, 0)$. The actual values of the corresponding position $X^a(t)$ and angle $\theta^a(t)$ are shown in figures 27(b) and (c), together with the noise fluctuations $\hat{n}_X^s$ and $\hat{n}_\theta^s$. Their comparison with the noise-free trajectories of the cart pole, as given by the solid lines of figure 16, shows the existence of small low-frequency oscillations around the equilibrium position, especially for θ.

In the second phase of collecting the training data, the values of the force $\vec{F}^a$ applied to the cart pole and the values of the state variables $\vec{Z}^s$ measured by the sensors were stored off-line through data communication links. Noise that may occur during the analog signal processing of the state vector $\vec{Z}^s$ was simulated by a four-dimensional vector $\hat{n}_{\vec{Z}}^{cs}$ normally distributed around zero

$$\vec{Z}^{\text{training}}(t) = \vec{Z}^s(t) + \hat{n}_{\vec{Z}}^{cs} \tag{29}$$

The noise that may be added to the force during the data processing was simulated by a normal distribution $\hat{n}_F^{ca}$

$$\vec{F}^{\text{training}}(t) = \vec{F}^a(t) + \hat{n}_F^{ca} \tag{30}$$

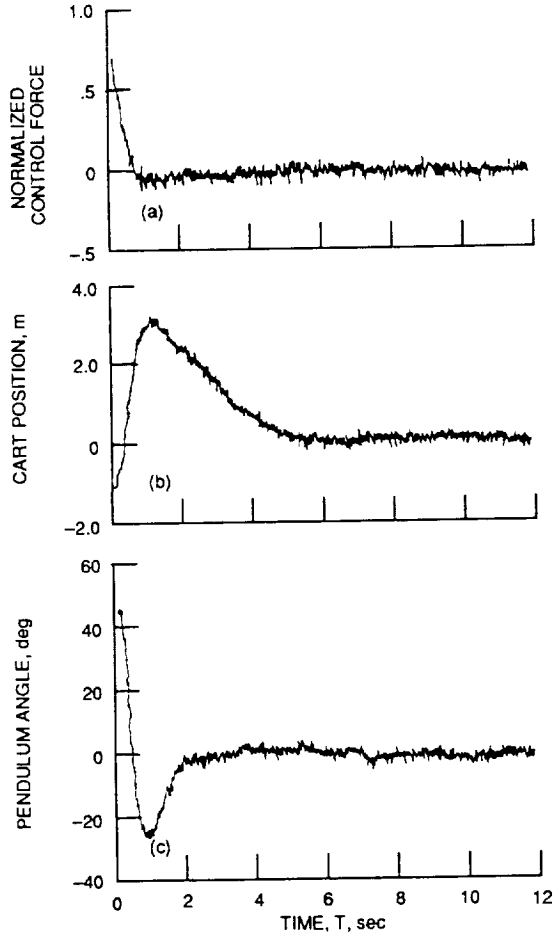
In this simulation, $N/S = 0.04$ was chosen for the noise fluctuations incurred during data processing: that is,

$$\frac{\sqrt{\text{var}(\hat{n}_F^{ca})}}{F_{\max}} = \frac{\sqrt{\text{var}(\hat{n}_{\vec{Z}}^{ca})}}{\max|\vec{Z}^a|} = 0.04$$

The effects of such measurement noise and data processing noise of the values of the force, $\vec{F}_{(t)}^{\text{training}}$, and state vector, $\vec{Z}_{(t)}^{\text{training}}$, used for training are illustrated in figure 28 for the initial condition $\vec{Z}(0) = (-1\text{ m}, 0, 45°, 0)$.

In the third phase, the noise-corrupted data were used to train the neural net according to the method developed in the previous section. After 1 million training iterations with $\alpha = 0.01$ and $\beta = 0$ (requiring 1 hr of VAX 8800 CPU time), the performance of the net was compared with the performance of the teacher. Figure 29(a) shows the force applied by the neural net controller to return the cart pole to its equilibrium position from the initial state $\vec{Z}(0) = (-1\text{ m}, 0, 45°, 0)$. For that motion, the values of the position $X^a(t)$ and angle $\theta^a(t)$, measured by the sensors, are shown in figure 29(b) and (c) together with the actual values $X^a(t)$ and $\theta^a(t)$. A comparison of figure 27(b) and (c) and with figure 29(b) and (c) shows that the motion of the cart pole was smoother, and more stable, when it was controlled by the net than when it was controlled by the teacher. These findings are illustrated further in figure 30 (teacher) and figure 31 (neural net) for the initial state vector $\vec{Z}(0) = (3\text{ m}, 0, -35°, 0)$. (Compare, in particular, figs. 30(c) and 31(c).)

In effect, the statistical nature of the procedure used to construct a training data sequence allows the net to overcome noise fluctuations, and to return the cart pole to its equilibrium position. A very interesting characteristic of the neurocontroller is the attenuation of the low-frequency fluctuations created by the presence of noise within the control loop.



(a) Normalized control force actually applied by the teacher-controller to the cart pole as function of time.

(b) Cart position as function of time.

(c) Pendulum angle as function of time.

Figure 27.—Dynamics of cart pole controlled by the nonlinear teacher in the architecture of figure 26 and operating in the presence of noise; initial state vector, $\vec{Z}(0) = (-1\text{ m}, 0, 45°, 0)$. Sensor noise-to-signal ratio, $(N/S)_{\text{sensors}}$, 0.2; control signal normalizing factor, F_0, 120 N.

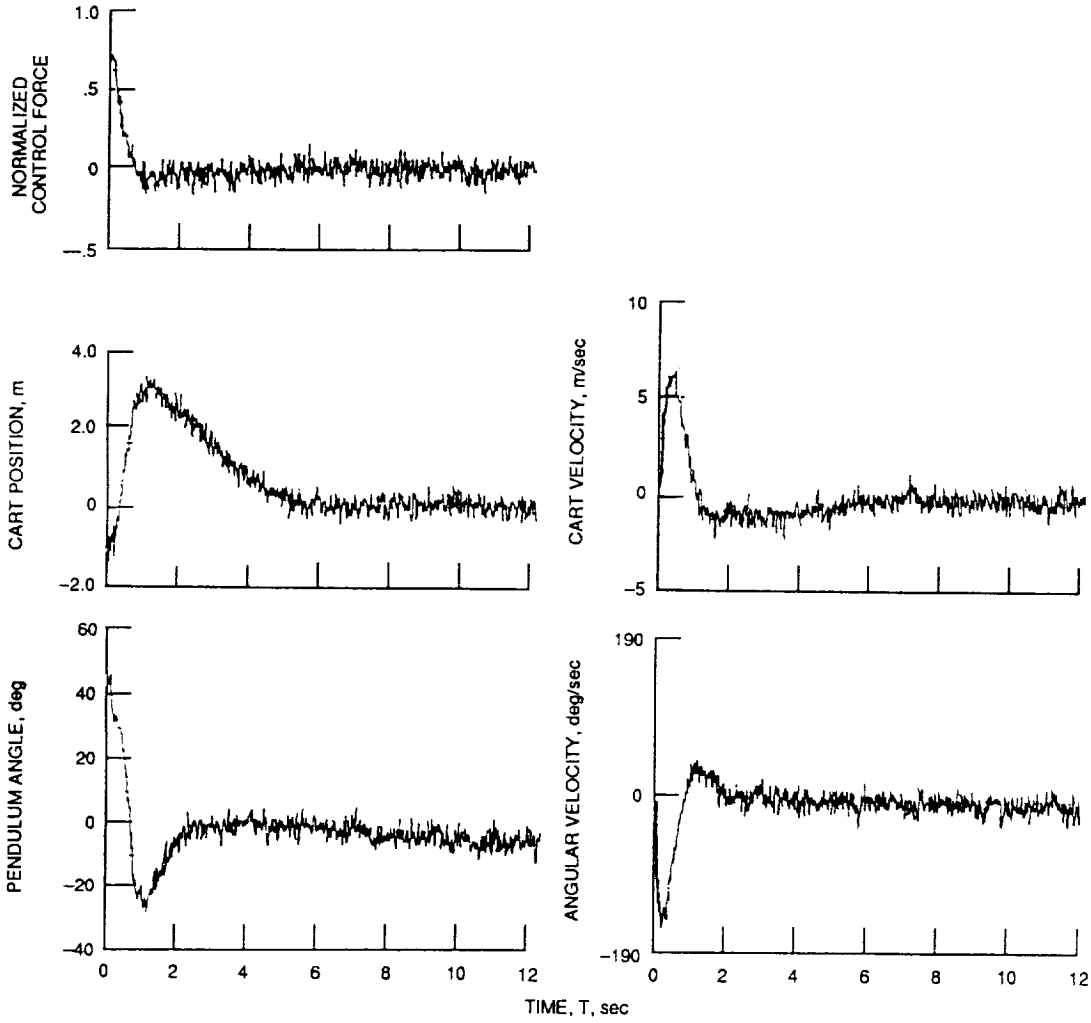
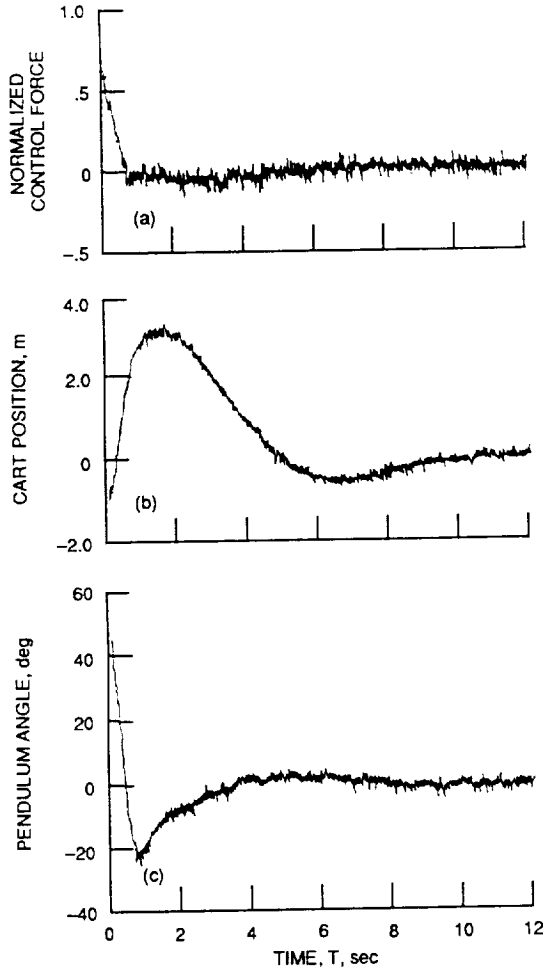
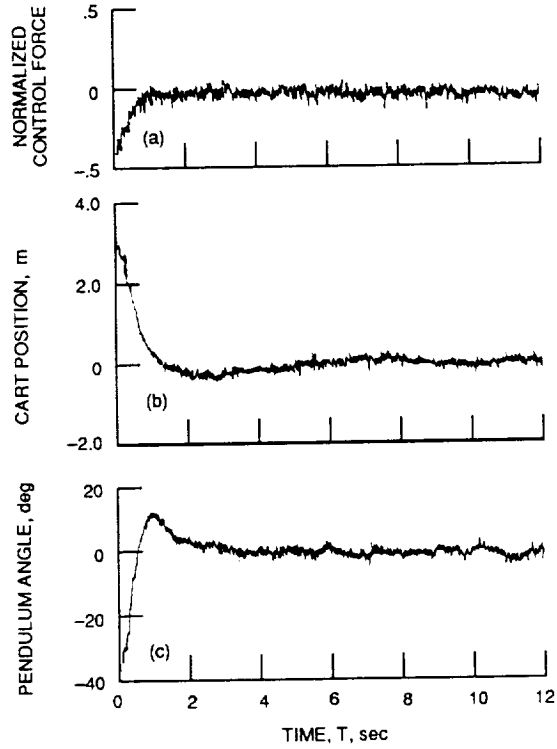


Figure 28.—Cart-pole dynamics sampled and transmitted to the four-layer neural network through the noisy data communications links of figure 26; processing noise-to-signal ratio, $(N/S)_{\text{processing}}$, 0.04.



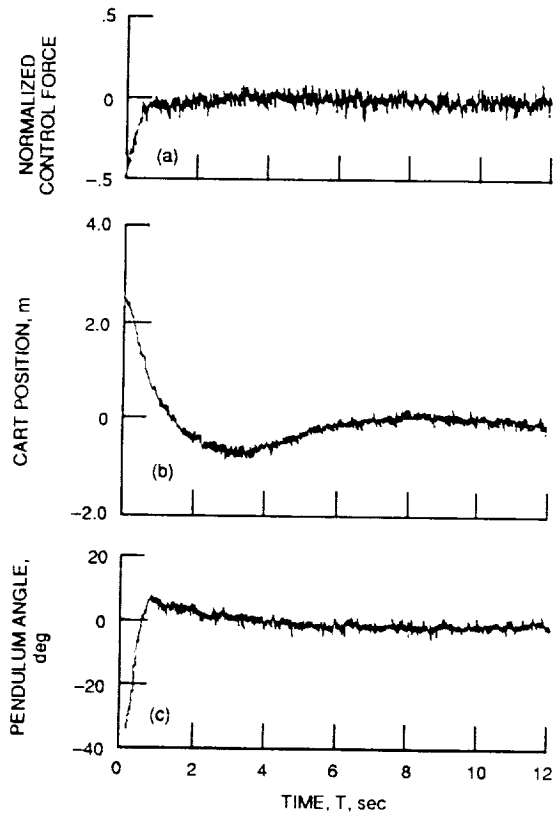
(a) Force to return cart pole to its equilibrium position from its initial state.
(b) Cart position as function of time.
(c) Pendulum angle as function of time.

Figure 29.—Dynamics of cart pole controlled by the neural network, trained from the noise-corrupted data in figure 26, and operating in the presence of noise after 1 million BEP iterations; initial state vector, $\vec{Z}(0)$ = (-1 m, 0, 45°, 0). Sensor noise-to-signal ratio, $(N/S)_{\text{sensors}}$, 0.02; control signal normalizing factor, F_0, 120 N; steepest descent parameter, α, 0.01; momentum coefficient, β, 0.



(a) Force to return cart pole to its equilibrium position from its initial state.
(b) Cart position as function of time.
(c) Pendulum angle as function of time.

Figure 30.—Dynamics of cart pole controlled by the teacher and operating in the presence of noise; initial state vector, $\vec{Z}(0)$ = (3 m, 0, −35°, 0). Noise-to-signal ratio, $(N/S)_{\text{sensors}}$, 0.02; control signal normalizing factor, F_0, 120 N.



(a) Force to return cart pole to its equilibrium position from its initial state.
(b) Cart position as function of time.
(c) Pendulum angle as function of time.

Figure 31.—Dynamics of cart pole controlled by the neural network, trained from the noise-corrupted data in figure 26, and operating in the presence of noise, after 1 million BEP iterations; initial state vector, $\vec{Z}(0)$ = (3 m, 0, −35°, 0). Noise-to-signal ratio, $(N/S)_{sensors}$, 0.02; control signal normalizing factor, F_0, 120 N; steepest descent parameter, α, 0.01; momentum coefficient, β, 0.

Conclusions

These results demonstrate that neural networks can satisfactorily extract a continuous-valued, nonlinear mapping between input and output data even when the training data are corrupted by measurement noise and data processing noise. In order to learn such a mapping (control law) on a certain region of the state space, the training data set must have the features that represent the mapping (control law) on the same region. In addition, the data of the training set must be ordered by magnitude to generate a training data sequence that is uniformly and randomly distributed throughout the state space of the original data. In the absence of noise, uniform distributions can be constructed by randomly sampling subintervals of the state space. However, the presence of noise mixes the populations of these subintervals, limiting the uniformity of the data distribution in the state space. Application of the variational principle to an error function determines the parameter values that minimize the effect of this noise contamination while maximizing the degree of embodiment of the mapping (control law) in the training data set.

Topics of further research would include the use of a time-dependent sampling rate and the use of neuromorphic classifiers in order to reduce the effect of the noise contamination and improve the neuromorphic learning of control laws. In the first case, decreasing the sampling rate as the controlled process returns to its equilibrium position would favor a more uniform probability distribution of the exact training data set, and thereby a more uniform distribution of the training sequence. In the second case, a neural architecture could be used as a noise-filtering preprocessor to selectively construct a training sequence that would be more uniformly distributed.

The neural computation was not only found to filter the noises and allow the neurocontroller to substitute for the teacher-controller, but also to provide a smoother mode of operation than the teacher. This means, for example, that a neural net could be trained off-line by observing a conventional digital microprocessor perform control operations in a very noisy environment and could be used as a substitute neurocontroller to improve the quality of the control in terms of stability. Besides the multiple advantages of the neural computation and of its practical implementation discussed in the introduction, the latter property may lead to new noise-filtering applications for neural network technology.

Acknowledgments

We express our gratitude to Carl Lorenzo for suggesting the analysis discussed in the section **Example of Neuromorphic Learning of Nonlinear Control With Measurement Noise and Data Processing Noise.** We are also grateful to Michelle Bright for suggesting the discussion of the control force normalization constant.

Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio, December 5, 1989

Appendix—Symbols

C	applied control signal
$D_{X\theta}$	domain of initial position and angle of the cart pole
e^2	total mean-squared error between targets and neural net outputs
e^2_{noise}	total mean-squared error between noisy targets and neural net outputs
$\bar{e}^2_{\text{noise}}$	normalized expression for e^2_{noise} going to zero in perfect learning
F	control force applied to cart
$F_{\max}$	range of variations of the control signal (force)
F_0	constant that normalizes the control signal over [-1, +1]
F^a	force applied to the cart pole by the actuator in the presence of noise
f_c	cutoff frequency above which the spectral components of the control signal and the state vector are small
f_s	sampling rate of control signal and state variables
$G(\vec{Z})$	variable of an error function
g	acceleration due to gravity
h_1,h_2,h_3,h_4	auxilliary expressions for the nonlinear dynamics of the cart pole
I_k	subinterval of [-1, +1]
i_{net}	total input signal of a neuron
k	coefficients of various parameters
L	distance between the base of the pole and the center of gravity of the pole
M	mass of cart
m	mass of pole
N_{F_n}	number of subintervals of the control force normalized in [-1, +1]
N_{motions}	number of observed responses of the cart pole to random displacements from its equilibrium position
$\hat{n}$	noises (independent, normally distributed, zero mean processes)
$O(\ )$	higher order terms
o_n	target output
$\bar{o}_n$	network output
p	p^{th} layer
S_0	entire region of the state space of the control law
S_1	region of the state space of the control law that is covered by the training data set
T	length of observation time of a cart-pole response
t_k	times at which samples are taken
u	normalized control force applied to cart (F/F_0)
var	variance
w	"synaptic" weight connecting two neurons
w_{th}	"synaptic" weight connecting a neuron to a neuron that is permanently "on" (defines its threshold)
X	axial location of the cart pole
$\dot{X}$	linear velocity of cart pole
$\ddot{X}$	linear acceleration of the cart
$\vec{Z}$	state vector of the controlled process
$\vec{Z}^s$	state vector measured by sensors
α	steepest descent parameter
β	momentum coefficient of the steepest descent
$\Delta_{i,p}$	signal error backpropagated to the i^{th} neuron of the p^{th} layer
ϵ	error between target and neural net output
η	force of friction acting on cart
θ	angle of pole displacement from the vertical
$\dot{\theta}$	angular velocity of the pole
$\ddot{\theta}$	angular acceleration of the pole
$\nu(p)$	number of neurons of the p^{th} layer
ϕ	exact mapping to be learned by the neural net

Subscripts:

ex	values that the state vector or control signal would have in the absence of noise (different from actual value due to the nonlinear effects of measurement noise)
i	i^{th} neuron
j	j^{th} neuron
k	value over the subinterval I_k
n^{th}	iteration number

Superscripts:

a	actual value
c	data communication
s	sensor measurement
target	value to be matched as closely as possible by the neural net
training	value used to train the network
—	normalized value

References

1. Merrill, W.C.; and Lorenzo, C.F.: A Reusable Rocket Engine Intelligent Control. AIAA Paper 88-3114, July 1988 (NASA TM-100963).
2. Lippman, R.P.: An Introduction to Computing with Neural Nets. IEEE ASSP Magazine, vol. 4, no. 2, Apr. 1987, pp. 4–22.
3. Hecht-Nielsen, R.: Theory of the Backpropagation Neural Network. International Joint Conference on Neural Networks, Vol. 1, IEEE, 1989, pp. 593–605.
4. Guez, A.; and Selinski, J.: A Trainable Neuromorphic Controller. J. Robotics Systems, vol. 5, no. 4, Aug. 1988, pp. 363–388.
5. Soucek, B.; and Soucek, M.: Neural and Massively Parallel Computers: The Sixth Generation. Wiley, New York, 1988.
6. DARPA Neural Network Study, AFCEA International Press, Fairfax, VA, 1988.
7. Widrow, B.: The Original Adaptive Neural Net Broom-Balancer. 1987 IEEE International Symposium on Circuits and Systems, Vol. 2, IEEE, 1987, pp. 351–357.
8. Barto, A.G.; Sutton, R.S.; and Anderson, C.W.: Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems. IEEE Trans. on Systems, Man and Cybernetics, vol. SMC-13, no. 5, Sept./Oct. 1983, pp. 834–846.
9. Rumelhart, D.E.; Hinton, G.E.; and Williams, R.J.: Learning Internal Representations by Error Propagation. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. I: Foundations, D.E. Rumelhart and J.L. McClelland, eds., MIT Press, 1986, pp. 318–362.
10. Guez, A.: Optimal Control of Robotic Manipulators. Ph.D. Thesis, Univ. of Florida, 1983.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-4176		

4. Title and Subtitle	5. Report Date
Neuromorphic Learning of Continuous-Valued Mappings From Noise-Corrupted Data Application to Real-Time Adaptive Control	May 1990
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Terry Troudet and Walter C. Merrill	E-4967
	10. Work Unit No.
	582-01-11
9. Performing Organization Name and Address	11. Contract or Grant No.
National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, D.C. 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

Terry Troudet, Sverdrup Technology, Inc., NASA Lewis Research Center Group, Cleveland, Ohio 44135.
Walter C. Merrill, NASA Lewis Research Center.

16. Abstract

The ability of feed-forward neural network architectures to learn continuous-valued mappings in the presence of noise was demonstrated in relation to parameter identification and real-time adaptive control applications. An error function was introduced to help optimize parameter values such as number of training iterations, observation time, sampling rate, and scaling of the control signal. The learning performance depended essentially on the degree of embodiment of the control law in the training data set and on the degree of uniformity of the probability distribution function of the data that are presented to the net during a training sequence. When a control law was corrupted by noise, the fluctuations of the training data biased the probability distribution function of the training data sequence. Only if the noise contamination is minimized and the degree of embodiment of the control law is maximized, can the neural net develop a good internal representation of the mapping and be used as a neurocontroller. A multilayer net was trained with back-error-propagation to control a cart-pole system for linear and nonlinear control laws in the presence of data processing noise and measurement noise. The neurocontroller exhibited noise-filtering properties and was found to operate more smoothly than the teacher in the presence of measurement noise.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Neural networks Adaptive control Parameter identification	Unclassified – Unlimited Subject Category 63

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price*
Unclassified	Unclassified	26	A03

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1990